Premium and Atrium using Unity Pro

Profibus DP bus

User Manual

10/2013

35006212.10

www.schneider-electric.com

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2013 Schneider Electric. All rights reserved.

Table of Contents

	Safety Information	**7**
	About the Book	**9**
Part I	**General introduction to communication on Profibus DP**	**11**
Chapter 1	**Introduction to Profibus DP**	**13**
	General introduction to Profibus DP	14
	General architecture and protocol for Profibus DP	15
	Multi-master architecture	17
	Features of Profibus DP	18
Part II	**Installation of Profibus DP hardware**	**19**
Chapter 2	**Performance**	**21**
	Data transfer capacity	22
	Network cycle	23
	Application response time	24
Chapter 3	**Description of the TSX PBY 100 module**	**27**
3.1	Description of module	28
	General description	29
	Operating mode	31
	Connecting the Profibus DP bus	32
3.2	Installing the module	33
	Mounting the module in a rack	33
3.3	Technical specifications	37
	Compatibility	38
	Standards and characteristics	39
	Operating conditions	40
Part III	**Software installation of the Profibus DP bus**	**43**
Chapter 4	**General**	**45**
	Principle	46
	Physical or logical addressing of inputs/outputs	47
	Mapping IW and QW addresses	48
Chapter 5	**TSX PBY 100 module configuration**	**51**
	Declaring the TSX PBY 100 module and accessing application screens	52
	Configuration screen for a Profibus DP link	53
	Data to be provided	55

Resulting data from the decoding of the *.CNF text file 56
Viewing Profibus DP master configuration 58
General module configuration 59
Module configuration file 61
Chapter 6 Programming Profibus DP communication 63
Profibus DP diagnostics 64
Diagnostics command 65
Examples of diagnostics command 67
Communication/operation report 69
Chapter 7 Debugging the TSX PBY 100 module 71
Description of the debug screen 72
Debugging Parameters 74
Chapter 8 TSX PBY 100 module diagnostics 77
Diagnostics of the module's status from the LEDs 78
Degraded project modes 79
Lists of diagnostics variables 81
List of available diagnostics 83
Compact diagnostics of all slaves 84
Slave diagnostics 85
General information on a slave 86
Slave configuration data 87
Typical errors 88
Chapter 9 Profibus DP communication language objects 91
9.1 Language objects and IODDTs for Profibus DP communication using the TSX PBY 100 module 92
Presentation of the language objects for Profibus DP communication 93
Implicit Exchange Language Objects Associated with the Application-Specific Function 94
Explicit Exchange Language Objects Associated with the Application-Specific Function 95
Management of Exchanges and Reports with Explicit Objects 97
9.2 General Language Objects and IODDTs for Communication Protocols 101
Details of IODDT Implicit Exchange Objects of Type T_COM_STS_GEN 102
Details of IODDT Explicit Exchange Objects of Type T_COM_STS_GEN 103

9.3 IODDTs for Profibus DP communication 105
Details of the implicit exchange objects of the T_COM_PBY-type IODDT 106
Details of the implicit exchange language objects for a Profibus DP function 110
Language objects associated with configuration 111
Error codes for module TSX PBY 100 112
9.4 The IODDT Type T_GEN_MOD Applicable to All Modules 114
Details of the Language Objects of the T_GEN_MOD-Type IODDT . . 114
Index 117

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This manual describes the implementation of hardware and software of the TSX PBY 100 module for Profibus DP communication with Premium and Atrium PLCs.

Validity Note

This documentation is valid from Unity Pro 8.0.

Product Related Information

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this product.

Follow all local and national safety codes and standards.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Part I
General introduction to communication on Profibus DP

Chapter 1
Introduction to Profibus DP

Subject of this Chapter

This chapter introduces the main features of communication on the Profibus DP.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
General introduction to Profibus DP	14
General architecture and protocol for Profibus DP	15
Multi-master architecture	17
Features of Profibus DP	18

General introduction to Profibus DP

Introduction

Profibus DP is a serial link field bus for sensors and actuators, which meets the requirements for use in an industrial environment.

This bus uses the master/slave process. The master subscriber manages and co-ordinates access to the bus, it transmits data to and receives data from all the subscribers.

Devices such as input/output modules are also available:

- compact Classic TIO slaves:
  - classic discrete inputs,
  - classic discrete outputs.
- DEA203 modular slaves
- Momentum modular slaves:
  - discrete inputs,
  - discrete outputs,
  - discrete inputs/outputs,
  - analog inputs/outputs.

Input/output modules

Input/output modules are used to link up sensors and actuators for checking or monitoring machines or processes to the Profibus DP system.

TSX PBY 100

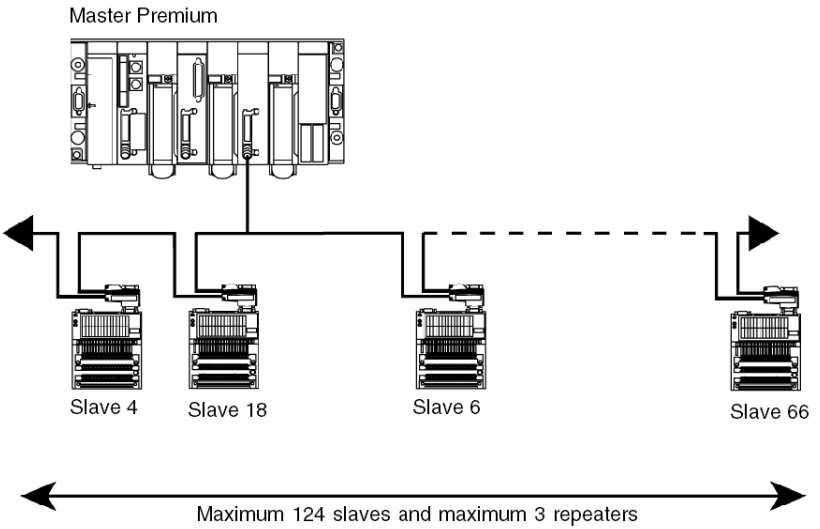
The module TSX PBY 100 *(see page 27)* enables Premium PLCs to be connected on the Profibus DP bus.

General architecture and protocol for Profibus DP

General architecture

The architecture of the Profibus DP field bus is used to implement the TSX PBY 100 module and slave devices.

This illustration shows the shortest network cycle time.



Connection rules

A Profibus DP bus can be made up of several electric and optic segments interconnected by repeaters.

Each of the electric segments must be adapted (impedance) and you must use:

- Two connectors: ref. 490NAD91103 (yellow) mounted on the devices situated at the ends of each electric segment.
- For the other connections, you must use these connectors: ref. 499NAD91104 or 490NAD91105 (gray).

You must ensure that there is cable screening continuity at connector level, otherwise the devices may be weakened.

It is advisable to use an optic segment between two constructions or to add surge absorbers on the electric segments.

Protocol

The protocol principle is based on a master/slave type bus. This principle guarantees excellent response times on I/O type exchanges (cyclic exchanges), with a maximum network cycle time less than 5 ms at 12 Mbds.

Only the master stations, sometimes called active stations, have access rights to the bus. The slave (or passive) stations can only respond to prompts and requests.

Several types of device are standardized:

- Master class 1 generally PLC, robot, digital command, etc.
- Master class 2 configuration devices, programming and master diagnostics.
- Slaves.

Profibus DP station addressing

Profibus DP stations can be identified by a number between 0 and 124 which defines the number of the station in the architecture (from 1 to 125).

This address corresponds to the station connection point on the bus registered in the configuration.

Multi-master architecture

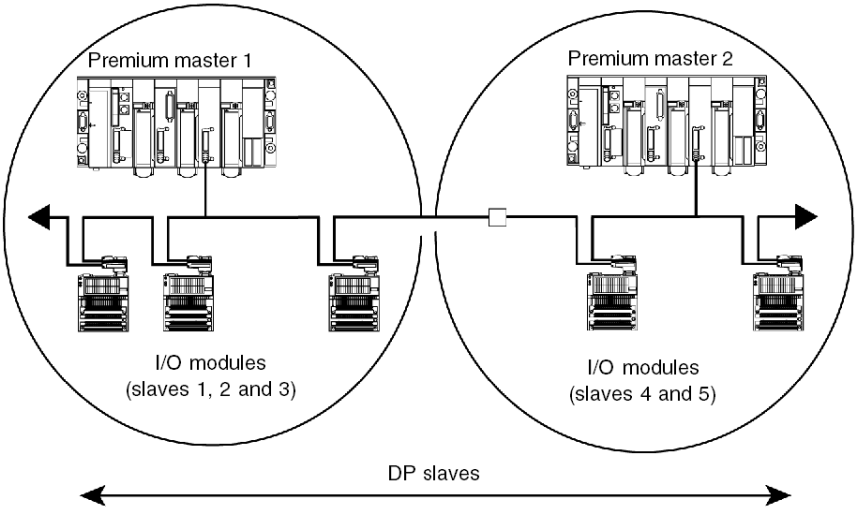
At a Glance

The Profibus DP field bus allows there to be several master stations.

In a multi-master configuration, each master station is associated to slaves and so forms a sub-system.

Illustration

This illustration describes a Profibus DP field bus multi-master architecture implementing a TSX PBY 100 module and slave devices.



Features of Profibus DP

Introduction

Profibus DP is a linear bus, designed for high speed data transfers. The PLC communicates with its peripheral devices via a high-speed serial link.

Data exchange is mainly cyclic.

Transmission features

This table describes the transmission features of the Profibus DP bus supported by the TSX PBY 100 module.

Topology	Linear bus with line terminators
Transmission mode	Half Duplex
Transmission rate	9.6 / 19.2 / 93.75 / 187.5 / 500 / 1500 Kbits/s up to 3 / 6 / 12 Mbit/s
Maximum length	100 m at 3 / 6 / 12 Mbit/s (400 m with 3 repeaters) 200 m at 1.5 Mbit/s (800 m with 3 repeaters) 500 m at 500 Kbit/s (2000 m with 3 repeaters) 1000 m at 187.5 Kbit/s (4000 m with 3 repeaters) 1200 m at 9.6 / 19.2 / 93.75 Kbit/s (4800 m with 3 repeaters)
Possible transmission media	Twisted pair line (standard version, type RS 485) Fiber optic link Waveguide
Connector	9-pin Sub-D

Capacity

This table describes the Profibus DP bus transmission capacity.

Number of master stations per PLC	0	TSX P57 104/154/1634
	1	TSX P57 204/254/2634/TSX PCI 57 204
	3	TSX P57 304/3634/354/PCI 57 354
	4	TSX P57 454/4634
	5	TSX P57 554/5634/6634
Number of slave stations	32 without repeaters	
Number of inputs/outputs	124 with maximum number of repeaters 2048 inputs / 2048 outputs maximum	
Number of repeaters	3	

Part II
Installation of Profibus DP hardware

Subject of this Part

This part presents the installation of Profibus DP hardware.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
2	Performance	21
3	Description of the TSX PBY 100 module	27

Chapter 2
Performance

Subject of this Chapter

This chapter introduces Profibus DP bus performance.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Data transfer capacity	22
Network cycle	23
Application response time	24

Data transfer capacity

Introduction

The TSX PBY 100 module requires slaves with configuration data of less than 250 bytes and diagnostics data of less than 244 bytes.

It is used to store configuration data for 125 devices whose total maximum size is 16 Kbytes.

Transmitted data

The following table shows the size of transferred input/output image data in words:

Data	minimum	maximum
Image of inputs in words (%IW) for configuration	-	242
Image of outputs in words (%QW) for configuration	-	242

Data per slave

The following table shows the size of data per slave in bytes:

Data	minimum	maximum
Configuration data per slave (in bytes)	31	250
Configuration data per slave	6	244
Maximum size of all configuration data	-	16 Kb

Network cycle

At a Glance

The network cycle depends on the rate of transfer, the number of slaves connected to the bus and the number of input/output words.

Configuration

The following table shows the network cycle times for several possible configurations.

Configuration	Network cycle time (ms)
Transfer rate 12 Mbit/s 124 slaves 242 input words and 242 output words	5 ms
Transfer rate 12 Mbit/s 124 slaves 126 input words and 126 output words	5 ms
Transfer rate 12 Mbit/s 32 slaves 32 input words and 32 output words	2.4 ms
Transfer rate 12 Mbit/s 1 slave 1 input word and 1 output word	1 ms
Transfer rate 500 Mbit/s 124 slaves 126 input words and 126 output words	100 ms
Transfer rate 500 Mbit/s 32 slaves 32 input words and 32 output words	25 ms
Transfer rate 500 Mbit/s 1 slave 1 input word and 1 output word	1.8 ms

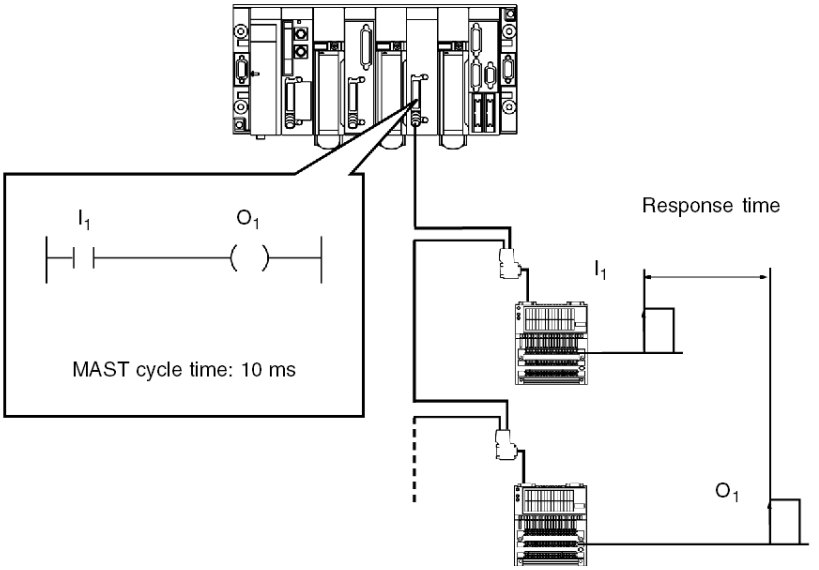
Application response time

At a Glance

The application response time is a logical response time, which does not take into account the filtering time or the response time of the sensor and actuator interfaces.

Illustration

The following example shows the time elapsed between acquiring an input and setting an output at a transmission rate of 12 Mbit/s.



Calculation example

The following table groups together the different elements for calculating the application response time.

max. updated %IW / QW	32		128		242	
Maximum number of I/O for the process	1024		4096		7744	
Maximum number of I/O modules	64		124		124	
	Min.	**Max. =** 2 x min.	**Min.**	**Max. =** 2 x min.	**Min.**	**Max. =** 2 x min.
Scanning time (ms) (acquiring image I_1)	2,44	4,8	5	10	11	22
MAST cycle time (in ms) (I_1 = O_1)	10,00	20,00	10,00	20,00	10,00	20,00
IBS scanning time (in ms) (updating O $image_1$)	2,44	4,8	5	10	11	22
Application response time (in ms)	**14,88**	**29,6**	**20**	**40**	**32**	**32**

Chapter 3
Description of the TSX PBY 100 module

Subject of this Chapter

This chapter introduces the main features of the TSX PBY 100 module.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
3.1	Description of module	28
3.2	Installing the module	33
3.3	Technical specifications	37

Section 3.1
Description of module

Subject of this Section

This section describes the physical appearance of the module and its operation.

What Is in This Section?

This section contains the following topics:

Topic	Page
General description	29
Operating mode	31
Connecting the Profibus DP bus	32

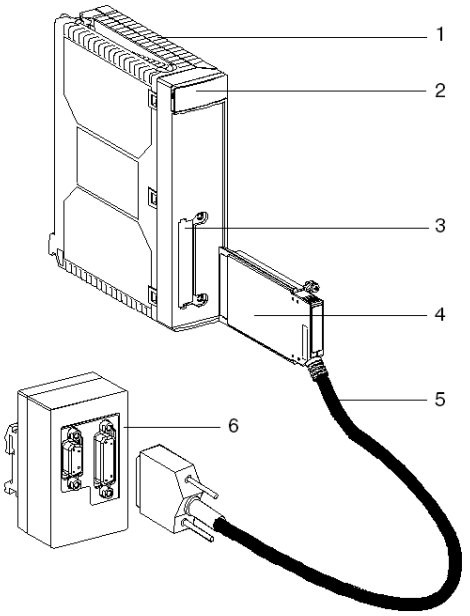
General description

At a Glance

The TSX PBY 100 module can be installed on a standard or extendable Premium PLC rack.

Illustration

The TSX PBY 100 module is made up of several elements:



Description of elements

The following table describes the different elements:

Number	Function
1	A host module which can be placed in any slot of a main or extension rack.
2	An indicator block made up of 4 LED *(see page 78)* indicators.
3	A slot for receiving a PCMCIA card.
4	A Profibus DP PCMCIA card.
5	A cable of 0.6m for linking up to a connection box.
6	A Profibus DPconnection box, Profibus DP bus connection interface.

This manual presents the **TSX PBY 100** module. This includes all the devices of which it is composed.

Services

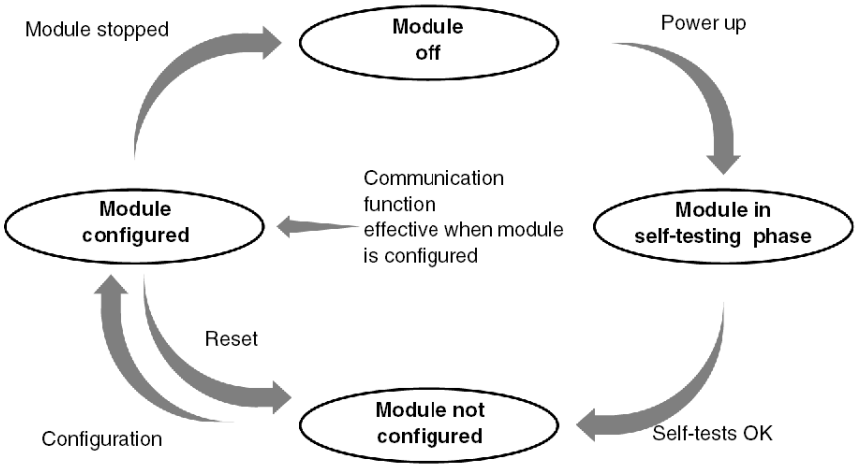
The TSX PBY 100 module is a master class 1 type device and provides the following services:

Services	Request or response	To or from	Comments
DATA_EXCHANGE	request	slave	input/output data transfer
SLAVE_DIAG	request	slave	slave diagnostics service
SET_PRM	request	slave	transmission of parameters to slaves on power-up
Chk_Cfg	request	slave	configuration check on power-up
GLOBAL_CONTROL	request	slave	global bus control (automatically carried out by the Profibus DPcard)
Get_Master_Diag	request	master class 2	master diagnostics service (automatically taken on by the Profibus DPcard)

Operating mode

Operation

The following illustration shows how the module operates:



Behavior

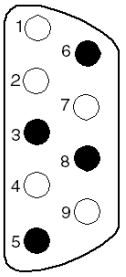
Description of the different operating modes of the module:

- **Profibus DP in RUN mode**: data exchange on the bus.
- **Task in RUN mode**: input/output update.
- **Task in STOP mode**:
  - input update,
  - output fallback strategy (maintained or reset to zero).

Connecting the Profibus DP bus

Illustration

Female 9-pin Sub-D RS 485 connector.



Description

Number	Description
1	Shield
2	M24: 24 V output voltage ground
3	**RxD/TxD-P**: positive data transmission (RD+ / TD+)
4	CNTR-P: positive repeater monitoring signal (direction monitoring): not used
5	**DGND**: data transmission ground
6	**VP**: line termination bias voltage
7	P24: output voltage 24 V
8	**RxD/TxD-N**: negative data transmission (RD- / TD-)
9	CNTR-N: negative repeater monitoring signal (direction monitoring): not used

NOTE: RxD/TxD-P, DGND, VP, RxD/TxD-N signals are mandatory. The other signals are optional.

Section 3.2
Installing the module

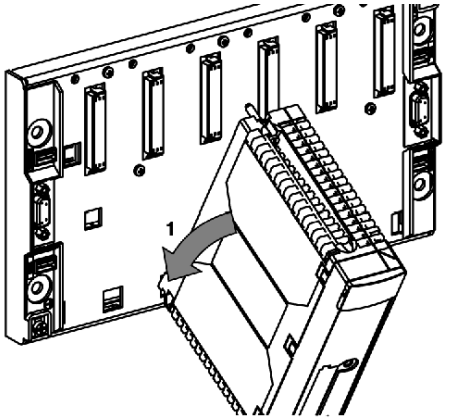
Mounting the module in a rack

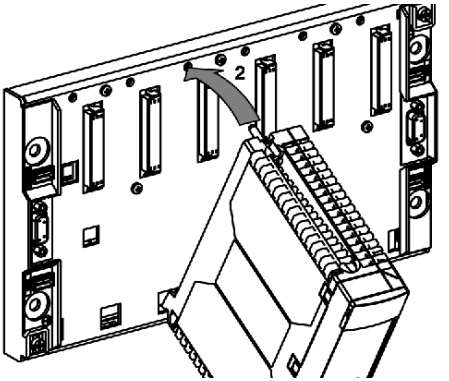
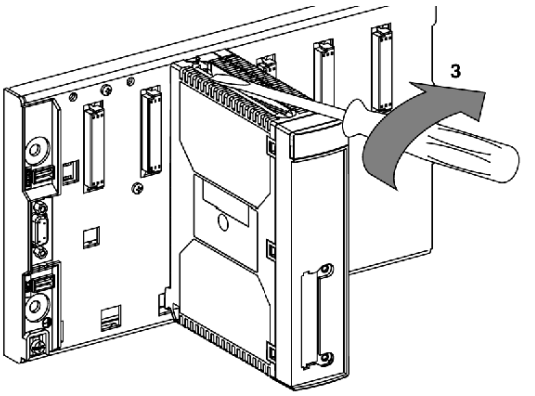
General

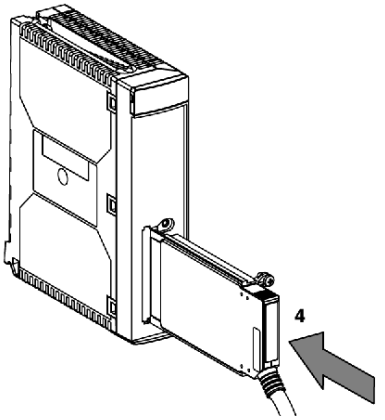
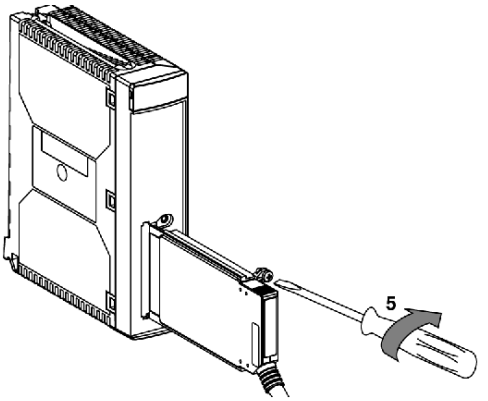
The mounting and removal of a module can be done with the power on.

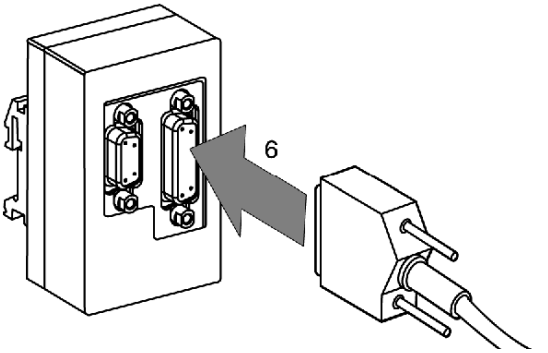
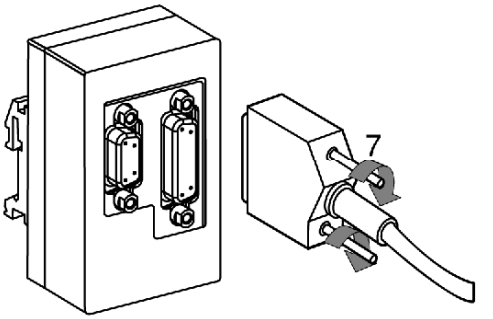
Inserting/extracting module with the power on must be done by doing/undoing the screws manually in order to maintain an adequate sequencing of the connection/disconnection of the signals on the X bus. Using an electric screwdriver cannot assure this sequencing.

Procedure

Step	Action	Illustration
1	Place the pins situated on the back of the module in the centering holes situated on the lower part of the rack.	

Step	Action	Illustration
2	Pivot the module in order to bring it into connect with the rack.	
3	Fix the module to the rack by tightening the screw situated on the upper part of the module	

Step	Action	Illustration
4	**You must not insert or take out the communication card while the reception module is powered up.**	
	Insert the card into its slot	
5	Screw the card in so that it cannot move, thus ensuring it will operate properly.	

Step	Action	Illustration
6	**You must not connect or disconnect the connection unit while the module is powered up.**	
	Connect the cable to the connection unit	
7	Screw the connector in so that it cannot move, thus ensuring a good connection.	

Section 3.3
Technical specifications

Subject of this Section

This section describes the technical specifications for using Profibus DP communication with the TSX PBY 100 module.

What Is in This Section?

This section contains the following topics:

Topic	Page
Compatibility	38
Standards and characteristics	39
Operating conditions	40

Compatibility

Hardware

Number of "application-specific" channels supported:

- Premium *(see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual)*
- Atrium *(see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual)*

The TSX PBY 100 host module is a master class 1 type device and can be integrated into a multi-master configuration. It is compatible with the following communication methods:

- master / slave,
- logical token ring.

Software

The TSX PBY 100 module is compatible with version ≥ V2.5.0.0 of the configuration software SyCon-PB and Unity Pro software.

These two software applications operate on Windows 2000 or XP.

Standards and characteristics

Standards

The TSX PBY 100 communication module complies with the following international standards:

EC Standards	IEC 1131-2, CENELEC (50081-2)
US Standards	UL508
Canadian Standards	CSA C22.2 No. 142-M1987

The TSX IBY 100 module also complies with the following standards:

Marine classification	• Germanischer Lloyd • Det Norsk Veritas • Bureau Veritas • Lloyds Register
US Standards	FM, Class I.Div.2 (CSA C22.2 No 213-M1987)

Certification

PBO

Characteristics

The electric characteristics are as follows:

- Logical DC V supply: 5 V DC provided by the rack power supply.
- Current consumed on 5 V: 400 mA.

Operating conditions

Operating temperature

- Ambient operating temperature: 0 °C to + 60 °C (IEC 1131-2 = + 5 °C to + 55 °C).

Hygrometry

- 30 % to 95 % (without condensation)

Altitude

- 0 to 2000 meters

Mechanical standards

- Vibration immunity: complies with the IEC 68-2-6 standard, Fc test.
- Shock immunity: complies with the IEC 68-2-27 standard, Ea test.

Electrostatic discharge standard

- Electrostatic discharge immunity: complies with the IEC 1000-4-2 standard, level 3.

NOTE: Minimum level in conditions defined by the standards

HF interference standard

- Immunity to radiated electromagnetic fields: complies with the IEC 1000-4-3 standard, level 3.
- Immunity to rapid burst transients: complies with the standard IEC 1000-4-4, level 3.
- Immunity to radiated electromagnetic fields: complies with the IEC 1000-4-12 standard, level 3.

NOTE: Minimum level in conditions defined by the standards

LF interference standard

- Complies with requirements of the IEC 1131-2 standard.

Premium PLC protection processing

Premium PLCs meet **"ACP"** (All Climate Processing) processing requirements.

For installations in industrial production workshops, or in an environment corresponding to **"PWH"** (Processing for Warm and Humid environments), Premium PLCs must be inserted into IP54 minimum protection envelopes as prescribed by IEC 664 and **NFC 20 040** standards.

Reminder

Premium PLCs have an IP20 protection rating. They can therefore be installed without an envelope in premises with restricted access which do not exceed pollution level 2 (control room with no machines or dust-producing activity).

NOTE: When a position is not occupied by a module, a TSX RKA 02 protection cover must be installed in it.

Requirements relating to transport and storage

These requirements comply with the IEC 1131-2 standard.

- Storage temperature: -25 degrees C to +70 degrees C.
- Relative humidity: 5 % to 95 % (without condensation).

Part III
Software installation of the Profibus DP bus

Subject of this Part

This part presents the software installation of Profibus DP using Unity Pro software.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
4	General	45
5	TSX PBY 100 module configuration	51
6	Programming Profibus DP communication	63
7	Debugging the TSX PBY 100 module	71
8	TSX PBY 100 module diagnostics	77
9	Profibus DP communication language objects	91

Chapter 4
General

Subject of this Chapter

This chapter provides a general introduction to installing the TSX PBY 100 module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Principle	46
Physical or logical addressing of inputs/outputs	47
Mapping IW and QW addresses	48

Principle

Introduction

When installing Profibus DP, the physical context of the project into which it will be integrated (rack, supply, processor, modules or devices, etc.) must be defined, and its software must be installed.

Installation of this software is carried out from the various Unity Pro editors:

- either in offline mode,
- or in online mode (modification is limited to certain parameters).

The sequence of installation phases defined below is recommended, though the order of certain phases may be modified (such as starting with the configuration phase).

Installation principle with a processor

The table below shows the different installation phases with a processor.

Phase	Description	Mode
Variable declaration	Declaration of the IODDT-type variables for application-specific modules and the project variables.	Offline (1)
Programming	Project programming.	Offline (1)
Configuration (2)	Module declaration.	Offline
	Configuration of the module channels.	
	Entering the configuration parameters.	
	Bus configuration declaration using SyCon-PB software and generation of a *.CNF text file.	
Association	Associating the IODDTs to the configured modules (variable editor).	Offline (1)
Generation	Project generation (analyzing and editing links).	Offline
Transfer	Transfer project to PLC.	Online
Adjustment/Debug	Debugging the project using the debug screens and animation tables of the SyCon-PB software.	Online
	Modification of the program and adjustment parameters.	
Documentation	Building the documentation file and printing the different data relating to the project.	Online (1)
Operation/Diagnostics	Displaying the different data needed for running the project.	Online
	Project / module diagnostics.	
Key		
(1)	These different phases may also be performed in the other mode.	
(2)	(1) When configuring a Profibus DP installation, SyCon-PB software needs to be used (available on CD-ROM, ref. TLX L FBC10M). This software comprises a library of profiles describing each device that can be connected to Profibus DP. For an update, consult our regional office.	

Physical or logical addressing of inputs/outputs

At a Glance

Inputs/outputs respect the topology used by the Unity Pro software and can be identified:

- either by physical addressing,
- or by logical addressing.

Topology

Addressing is defined in the following way:

%	I or Q	X, W or D	r	.	m	.	c	.	d	.	j	.
Symbol	Type of objects I = input Q = output	Format X = Boolean W = word D = double word	Rack address r = 0 to 7		Module position y = 0 to 14		Channel no. c = 0		Position r = 0 to 253		bit j = 0 to 15	

Block assignment

DP data is exchanged in the form of input/output blocks. All slave input data is indexed by adjacent %IW blocks. Slave output data is indexed by adjacent %QW blocks. The continuity of %IW and %QW blocks is valid even for a modular slave.

Each data block for a slave starts with a new %IW or %QW. As a result, the first I/O word of a slave is always associated to a new %IW or %QW.

In the event where a slave image (%IW or %QW) has a special size (for example 1 byte or 3 bytes), it is completed by unused bytes in order to manipulate the I/O words.

Example

The table below describes an example of assignment:

Input image				
Slave 2 2 words		Slave 1 1 byte	Unused byte	Slave 17 1 word
Output image				
Slave 17 2 words		Slave 2 1 byte	Unused byte	
%QWr.m.0.d	%QWr.m.0.d+1	%QWr.m.0.d+2 Only bits 0 to 7 are significant		

Mapping IW and QW addresses

General

Mapping input/output data addresses is used to achieve the clearest possible addressing.

A slave can be made up of several modules of different data sizes. In this case, misalignment of addresses can happen.

To avoid this, modules can be physically positioned in the slave rack by:

- grouping together input modules of a particular size (e.g.: 1 byte) for each pair,
- grouping together output modules of a particular size (e.g.: 1 byte) for each pair,
- positioning a single input module of a particular size (e.g.: 1 byte) at the last input module position,
- positioning a single output module of a particular size (e.g.: 1 byte) at the last output module position.

Example: non-mapped modules

Slave x in non-mapped modules

Module A 1 input word	**Module B** **1 input byte**	**Module C** **1 output byte**	Module D 1 input word	Module E 1 output word	Module F 1 output byte

Input image

%IWr.m.0.d	%IWr.m.0.d+1	%IWr.m.0.d+2	
Module A 1 input word	**Module B** **1 input byte**	Module D 1 input word	Unused byte

Output image

%QWr.m.0.d	%QWr.m.0.d+1	
Module C **1 output byte**	Module E 1 output word	Module F 1 output byte

Example: mapped modules

Slave x in mapped modules

Module A 1 input word	Module D 1 input word	**Module B** **1 input byte**	Module E 1 output word	**Module C** **1 output byte**	Module F 1 output byte

Input image

%IWr.m.0.d	%IWr.m.0.d+1	%IWr.m.0.d+2	
Module A 1 input word	Module D 1 input word	**Module B** **1 input byte**	Unused byte

Output image

%QWr.m.0.d	%QWr.m.0.d+1	
Module E 1 output word	**Module C** **1 output byte**	Module F 1 output byte

Chapter 5
TSX PBY 100 module configuration

Subject of this Chapter

This chapter describes the different configuration options of the TSX PBY 100 module.

What Is in This Chapter?

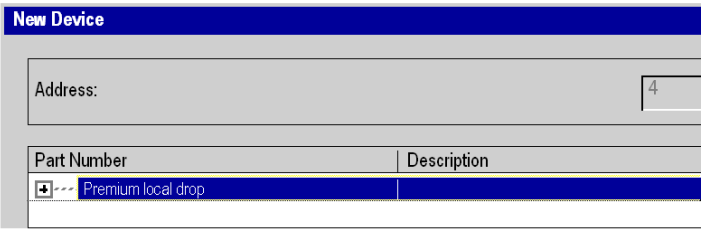
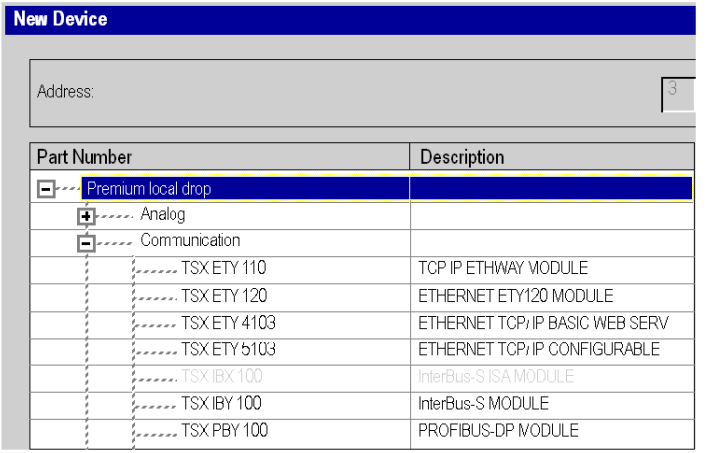
This chapter contains the following topics:

Topic	Page
Declaring the TSX PBY 100 module and accessing application screens	52
Configuration screen for a Profibus DP link	53
Data to be provided	55
Resulting data from the decoding of the *.CNF text file	56
Viewing Profibus DP master configuration	58
General module configuration	59
Module configuration file	61

Declaring the TSX PBY 100 module and accessing application screens

How to access the link

The following table shows the procedure for accessing the Profibus DP link:

Step	Action
1	Open the hardware configuration editor.
2	Select the slot where you wish to insert the module.
3	Select the **New Device** command from the contextual menu. **Result**: the **New Device** appears. 
4	Develop the **Premium local drop** line and the **Communication** line by clicking on the + sign. **Result**: 
5	Select the **TSX PBY 100** module then validate by clicking **OK**.
6	In the **X Bus** window, select the **TSX PBY 100** module.
7	Select the **Open Module** command from the contextual menu. **Result:** the configuration screen of the module appears.

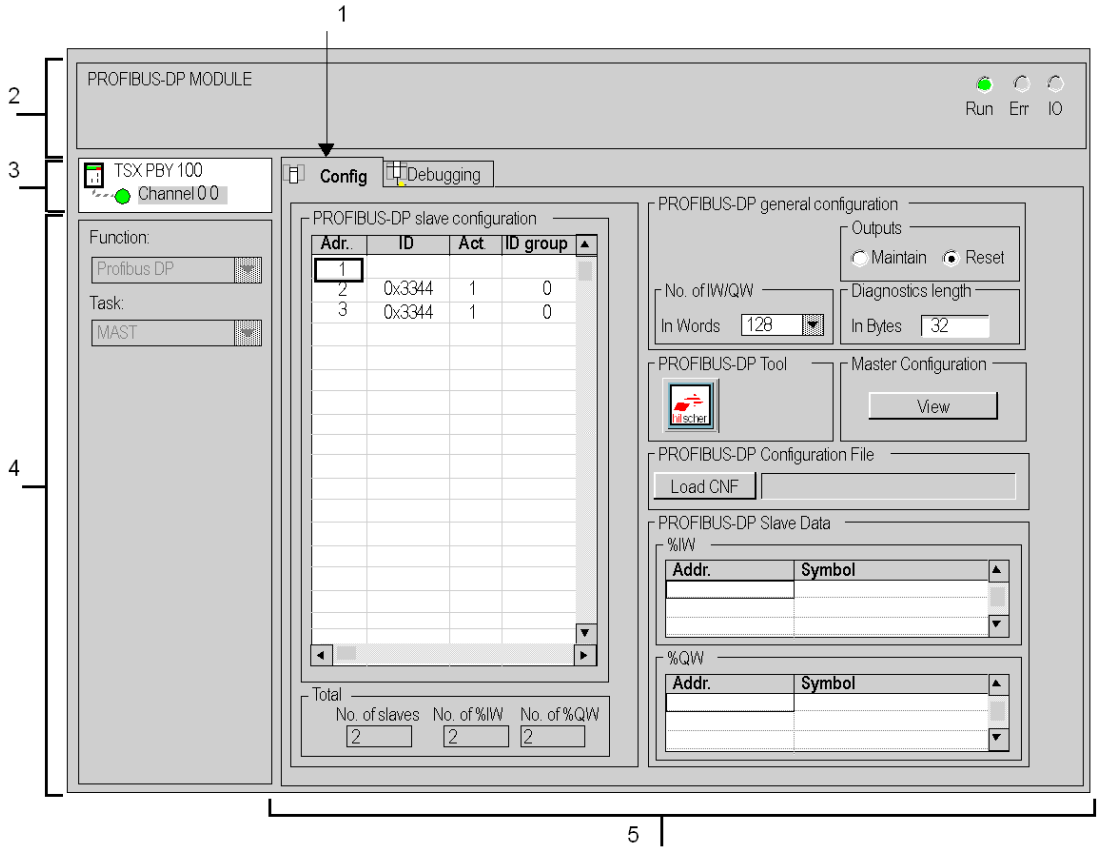
Configuration screen for a Profibus DP link

At a Glance

The configuration screen is made up of different areas and is used to set the features required for a Profibus DP link

Illustration

The figure below represents a configuration screen.



Description

The following table shows the different elements of the configuration screen and their functions.

Number	Element	Function
1	Tabs	The tab in the foreground indicates the mode in progress (**Configuration** in this example). You can select each mode by clicking the corresponding tab. The available modes are: ● **Configuration**, ● **Debugging**, accessible only in Online mode,
2	**Module** area	This provides a reminder of the module's abbreviated title and uses LEDs to indicate the module status in online mode.
3	**Channel** area	Is used: ● By clicking on the reference number, to display the tabs: ● **Description** which gives the characteristics of the device. ● **I/O Objects** *(see Unity Pro, Operating Modes)* which is used to presymbolize the input/output objects. ● **Fault** which shows the device faults (in online mode). ● To select the channel, ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** area	This lets you choose the general parameters associated with the channel: ● **Function:** the Profibus DP function is frozen (grayed out). ● **Task:** defines the **MAST** or **FAST** task through which the channel's implicit exchange objects will be exchanged.
5	**Configuration** area	This lets you configure the channel configuration parameters. Certain choices may be frozen and appear in gray. It is divided into five areas: ● the Profibus DP bus configuration *(see page 56)* area, ● the SyCon software launch and configuration *(see page 59)* file selection area, ● the Profibus DP bus general parameters *(see page 55)* area, ● the area giving access to "**Viewing**" *(see page 58)* of the parameters of the master and the Profibus DP bus, ● the area showing the input data *(see page 57)* and output data associated with a device.

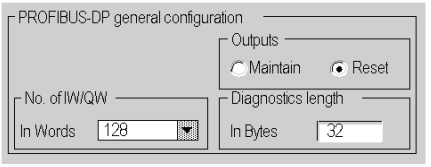
Data to be provided

At a Glance

To configure the communication channel, you must complete the parameters in the **General parameters** area dedicated to the project:

General parameters

The area looks like this:



- The **Number of IW/QW** field is used to select the number of words used for the inputs/outputs: 32, 64, 128 or 242.
- The **Outputs** field is used to select the fallback mode of the outputs:
  - **Maintain**: the value of the outputs is maintained.
  - **Reset**: reset to zero.
- The **Diagnostic Length** field is used to select the diagnostics length in bytes from 6 to 244 bytes (32 by default). The size configured should be sufficient to contain the most important bus diagnostics. If the size is insufficient, the slave concerned will not be active on the bus because its diagnostics will be invalid.

NOTE: In order to optimize performance, select a minimum number of input/output words and diagnostics bytes compatible with the actual bus configuration.

Resulting data from the decoding of the *.CNF text file

At a Glance

One part of the configuration screen is used to display the Profibus DP field bus topology as well as information on the slaves associated with the module.

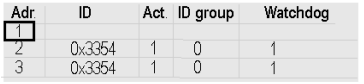
These are split into three areas:

- the **Profibus DP slave configuration** area,
- the **Total input/output data** area,
- the **Profibus DP slave data** area.

Profibus DP configuration

The **Profibus DP slave configuration** drop-down list shows the configuration of the Profibus DP field bus. It shows the contents of the selected *.CNF text file. The configuration of the 125 possible devices can be accessed in this manner.

Each line of this drop-down list shows the status of a single device. A line is presented in the format:

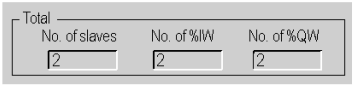


Adr	ID	Act	ID group	Watchdog
1				
2	0x3354	1	0	1
3	0x3354	1	0	1

- The **Adr** field indicates the address of the slave device (between 1 and 125).
- The **ID** field indicates an ID code (identification number supplied by the manufacturer).
- The **Act** field shows whether the slave is configured and present on the bus (1 configured and present).
- The **ID group** field shows whether the slave is made up of several modules.
- The **WatchDog** field shows the state of the watchdog.

Total input/output data

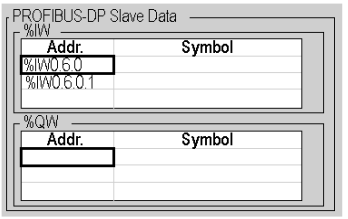
The area looks like this:



- The first field indicates the total number of slaves,
- The second field indicates the total number of input words,
- The third field indicates the total number of output words,

Profibus DP slave data

The area looks like this:



Two lists displaying the input/output addresses and symbols:

- the **%IW** list shows the input data relating to the selected device, with their associated symbol,
- the **%QW** list shows the output data relating to the selected device, with their associated symbol,

Viewing Profibus DP master configuration

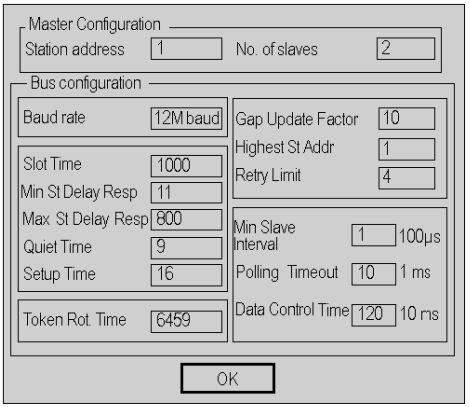
At a Glance

By pressing the **View** button, you are able to view the master and bus configuration parameters. This screen is enabled when you have selected a *.CNF text file.

If no *.CNF text file has been selected, a default file appears. It shows a master module with no slave.

Illustration

The screen is presented in the following format:



NOTE: For further information, please refer to the SyCon-PB software documentation and the Module configuration file *(see page 61)*.

General module configuration

At a Glance

Module configuration is split into two parts:

- Configuring the general parameters.
- Configuring the TSX PBY 100 module.

How to configure general parameters

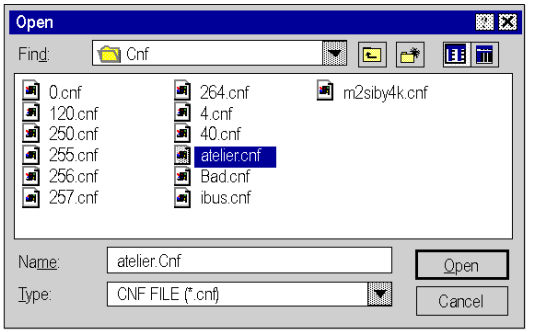
The following procedure is used to configure the general parameters.

Step	Action
1	Select the type of task that will drive the bus.
2	Select the number of words used for the inputs/outputs.
3	Select the action of PMS services upon the application stopping.
4	Select the action of output upon the application stopping.

How to configure the PBY module

The following procedure is used to configure the TSX PBY 100 module with SyCon-PB.

Step	Action
1	Click on the **Hilscher** button. **Result:** SyCon-PB software is activated.
2	Under SyCon-PB, configure: ● the bus topology, ● memory allocation: addressing for each image module in the %IW and %QW registers, ● group settings, ● special functions.
3	Export this configuration into the *.CNF text file.

Step	Action
4	Click on the **Load CNF** button. **Result:** the following window appears. 
5	Find and select the *.CNF text file which describes the configuration being used.
6	Confirm your selection using the **Open** button. The file is rejected if: ● the file format is incorrect, ● there are over 125 devices.
7	Confirm the configuration.

Module configuration file

At a Glance

A file describing the project configuration for the TSX PBY 100 module is available in the Unity Pro documentation editor.

Illustration

It is presented in the following format:

	2: MODULE Profibus DP		
Module identification			
Product Réf.:	TSX PBY 100	**Designation :**	Profibus DP module
Address:	0.2	**Symbol :**	
Title of the channel			
Profibus DP Specific fonction:			
Task:	MAST		
Event:			
Profibus DP general configuration:			
Outputs:	Reset		
No. of IW/QW:	128 words	Diagnostics length:	32 bytes
Profibus DP configuration file:			
Profibus DP master configuration:			
Station address:	1	No. of slaves:	0
Baud rate:	1.5M baud		
Slot time:	2000 tBit	Quiet Time:	6 tBit
Min St Delay Resp:	11 tBit	Max St Delay Resp:	55 tBit
Setup Time:	1 tBit	Token Rot. Time:	50000 tBit
Gap Update Factor:	1	Retry Limi :	3
Highest St Addr::	126	Min Slave Interval:	1 * 100 microsecondes
Polling Timeout:	500 ms	Data Control Time:	100 * 10ms
Profibus DP slave configuration			
Profibus DP slave langage objects			

Key:

Slot time	Maximum waiting time before the master starts responding to a request.
Min St Delay Resp.	Minimum waiting time before a slave can reply (transmission delay included).
Setup Time	Waiting time between the sending of the last bit by the slave and the master's acceptance of the response.
Gap Update Factor	Number of tokens the master uses to search for other masters on the network (for example: 10 = every 10 tokens).
Highest St addr.	The master looks for the other masters on the network masters only as far as this address. Not supported by the TSX PBY 100 module.
Polling timeout	Is only significant in exchanges between two masters. Not supported by the TSX PBY 100 module
Quiet time	Time needed for a signal to return to zero after sending a frame. During this time no devices are active on the bus.
Max St Delay Resp.	Maximum time during which the master waits for the slave to respond.
Token Rot. Time	Maximum token rotation time.
Retry Limit	When a slave does not respond, the master again asks the slave to transmit. It scans the next slave when the Retry Limit delay has been reached.
Min Slave Interval	Minimum time before a slave is re-polled.
Data Control Time	Maximum time for data exchange between the master and each slave.

Chapter 6
Programming Profibus DP communication

Subject of this Chapter

This chapter describes the programming component in the installing of Profibus DP communication.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Profibus DP diagnostics	64
Diagnostics command	65
Examples of diagnostics command	67
Communication/operation report	69

Profibus DP diagnostics

General

Profibus DP diagnostic functions are used to quickly find and identify faults on devices connected to the bus. Diagnostic messages are exchanged on Profibus DP via the TSX PBY 100 master module.

There are four types of diagnostics:

- **Master diag**: complete diagnostics on the TSX PBY 100 master module.
- **Slave diag**: complete diagnostics on a single slave.
- **Compressed diag** : compressed diagnostics on all slaves.
- List of diagnostics available for each slave.

Each of these diagnostics can be read by the Unity Pro software or by any other debugging PC.

Diagnostics command

At a Glance

A diagnostics command is sent by the SEND_REQ function block.

The SEND_REQ function must be used for reading or resetting the various diagnostics counters.

In order to handle common problems, the TSX PBY 100 module provides four diagnostics counters per slave *(see page 86)*. These counters can be accessed via the SEND_REQ function and are presented in the form of a byte table.

Syntax

The communication function syntax is presented in the following format:

```
SEND_REQ (ADDR('r.m.c'), 16#0031, %MWi:3, % MWk:4, %MWj:L)
```

The following table describes the various parameters of the function:

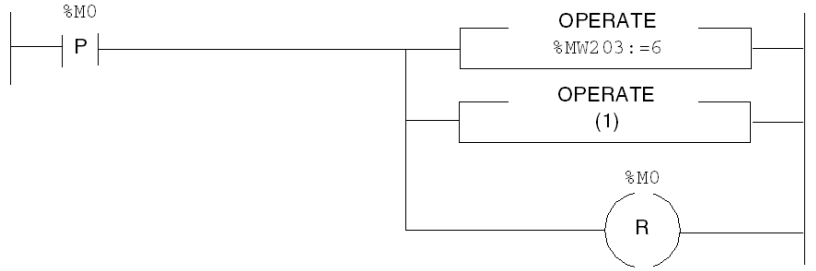
Parameter	Description	
ADDR('r.m.c')	**r**	Rack number
	m	Module number
	c	Channel number (0 for Profibus DP)
16#0031	Request code	

Parameter	Description		
%MWi:3	Request parameters		
	%MWi	Type of diagnostics	
	%MWi.0....8	0...125 126 127 128 129 130 131 132	Diagnostics on slave x List of diagnostics available Compressed diagnostics on all slaves Diagnostics on master module Total diagnostics counters Faulty exchange counters Downtime counters Invalid response counters
	%MWi.9	Reserved	
	%MWi.10	Reserved	
	%MWi.11	Reserved	
	%MWi.12	If activated, reading of configuration data for slave selected by %MWi.0...8 (= 0 to 124)	
	%MWi.13	If activated, reading of information for slave selected by %MWi.0....8 = 0 to 124	
	%MWi.14	If activated, resetting the list of available diagnostics or the counters specified by %MWi.0....8 = 126, 129 to 132	
	%MWi.15	If activated, reading of the list of available diagnostics or the counters specified by %MWi.0....8 = 126, 129 to 132	
	%MWi+1	Start address in the diagnostics table (default value is 0). To access part of the diagnostics table, it is possible to specify a start word in the table (Offset start)	
	%MWi+2	Length of diagnostics to be read	
%MWk:4	Management table for the function SEND_REQ		
%MWj:L	Reception table of length **L**, starting at the word **%MWj** which contains the diagnostics information. The responses obtained depend on the type of diagnostics performed. For further details on the diagnostics information that cab be obtained using the SEND_REQ communication function, consult the diagnostics chapter *(see page 77)*.		

Examples of diagnostics command

Reading diagnostics words from a slave

Diagnostics carried out on slave 2.



(1) `SEND_REQ(ADDR('0.6.0'),16#0031,%MW100:3,%MW200:4,%MW104:32)`

The table below describes the parameters:

Parameters	Variables	Values
Address	-	ADDR('0.6.0')
Request code	-	16#0031
Data to be sent	%MW100:3	2 (slave address in decimal) 0 (diagnostics table address in decimal) 32 (length of diagnostics table in decimal)
Exchange	%MW200:4	-
Reception Zone	%MW104:32	-

Diagnostics on a master

`SEND_REQ(ADDR('0.6.0'),16#0031,%MW100:3,%MW200:4,%MW104:32)`

Parameters	Variables	Values
Address	-	ADDR('0.6.0')
Request code	-	16#0031
Data to be sent	%MW100:3	126 (master code in decimal) 0 (diagnostics table address in decimal) 32 (length of diagnostics table in decimal)
Exchange	%MW200:4	-
Reception Zone	%MW104:32	-

Resetting the diagnostics counter

```
SEND_REQ(ADDR('0.6.0'),16#0031,%MW100:3,%MW200:4,%MW104:32)
```

Parameters	Variables	Values
Address	-	ADDR('0.6.0')
Request code	-	16#0031
Data to be sent	%MW100:3	16#4081 (initialization of the total diagnostics counter in hexadecimal) 0 (diagnostics table address in decimal) 32 (length of diagnostics table in decimal)
Exchange	%MW200:4	-
Reception Zone	%MW104:32	-

Communication/operation report

Description

These messages are common to all types of requests.

Communication report (least significant byte)		
Value	**Meaning**	
16#00	Correct exchange	
	Operation report (most significant byte)	
	Value / error code	**Meaning**
	Send request code in increments of 16#30	Positive result
	16#01	Request not processed
	16#02	Incorrect response
	16#03	Reserved
16#01	Exchange stopped on timeout	
16#02	Exchange stopped on user request (CANCEL)	
16#03	Incorrect address format	
16#04	Incorrect target address	
16#05	Incorrect management parameter format	
16#06	Incorrect specific parameters	
16#07	Problem with sending to destination device	
16#08	Reserved	
16#09	Size of receive buffer is insufficient	
16#0A	Size of send buffer is insufficient	
16#0B	No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor.	
16#0C	Incorrect exchange number	
16#0D	No telegram received	
16#0E	Incorrect length	
16#0F	Telegram service not configured	
16#10	Network module missing	
16#FF	Message refused	
	Operation report (most significant byte)	

	Value / error code	Meaning
	16#01	Lack of resources communicating with the processor
	16#02	Lack of line resources
	16#03	Device missing
	16#04	Line error
	16#05	Length error
	16#06	Communication channel fault
	16#07	Addressing errors
	16#08	Application fault
	16#0B	No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor.
	16#0D	Destination missing
	16#0F	Intra-station routing problem or channel not configured
	16#11	Address format not handled
	16#12	Lack of destination resources
	16#FD	Invalid parameter

Chapter 7
Debugging the TSX PBY 100 module

Subject of this Chapter

This chapter describes the different debug options of the TSX PBY 100 module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Description of the debug screen	72
Debugging Parameters	74

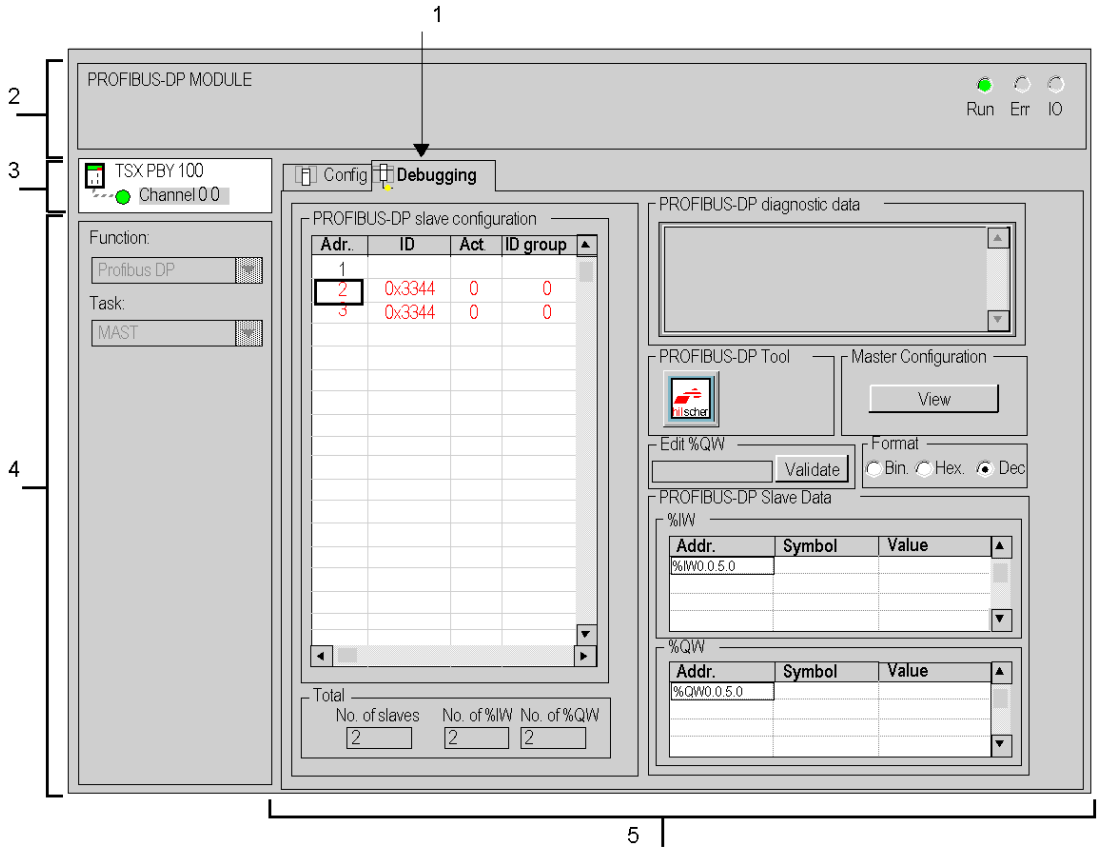
Description of the debug screen

At a Glance

The Debugging function or the ability to double click on the TSX PBY 100 graphical module in the Unity Pro configuration is only available in online mode.

Illustration

The figure below shows an example of a debug screen dedicated to Profibus DP communication.



Description

The table below shows the different elements of the debug screen and their functions:

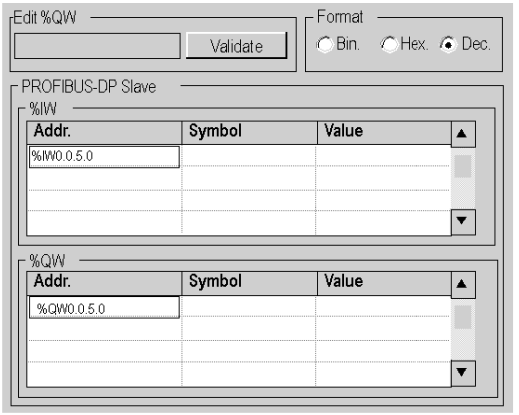
Number	Element	Function
1	Tabs	The tab in the foreground indicates the mode in progress (**Debug** in this example). You can select each mode by clicking the corresponding tab. The available modes are: ● **Debug** which can be accessed only in online mode. ● **Configuration**.
2	**Module** area	Provides a short reminder title of the module. In the same area there are 3 LEDs which indicate the status of the module in online mode: ● **RUN** indicates the operating status of the module, ● **ERR** indicates an internal fault in the module, ● **I/O** indicates a fault from outside the module or an application fault.
3	**Channel** area	Is used: ● By clicking on the reference number, to display the tabs: ● **Description** which gives the characteristics of the device. ● **I/O Objects** *(see Unity Pro, Operating Modes)* which is used to presymbolize the input/output objects. ● **Fault** which shows the device faults (in online mode). ● To select the channel, ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** area	Provides a reminder of the communication channel settings: ● **Function:** provides a reminder of the configured communication function. This heading is frozen. ● **Task:** specifies the **MAST** or **FAST** task configured. This heading is frozen.
5	**Viewing and control** area	Used to debug the channel. Certain choices may be frozen and appear in gray. It is divided into five areas: ● the Profibus DP bus configuration *(see page 56)* area, for when a device contains a fault: ● the cursor places itself over that device, ● the corresponding line appears in red. ● running the SyCon software, ● the diagnostics *(see page 64)* data of the Profibus DP bus, ● the area giving access to "**Viewing**" *(see page 58)* of the parameters of the master and the Profibus DP bus, ● the area showing the input data *(see page 74)* and output data associated with a device.

NOTE: all unavailable LEDs and commands appear in gray.

Debugging Parameters

Slave Data

To display I/O data values for a device, select **Profibus DP slave configuration** from the drop-down list.



Two pull down lists show the input/output data values:

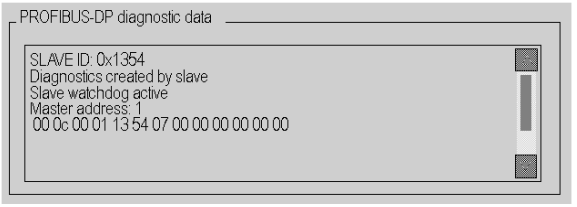
- The **%IW** area field displays the input data list for the device selected, with the symbol and the associated value for each data item.
- The **%QW** area field displays the output data list for the device selected, with the symbol and the associated value for each data item.
- The field of the **Edit %QW** area is used to enter the value of a %QW data item.
- The **Format** area field indicates the type of display for each data item:
  - hexadecimal,
  - decimal,
  - ASCII.

NOTE: Forcing is not authorized for %IW and %QW language objects.

The PLC switching to STOP mode makes the module's fallback values appear in red.

Diagnostics window

This window shows all the diagnostics information for a device. By selecting a device from the **Profibus DP slave configuration** list, its diagnostics appear in the **Profibus DP diagnostic data** window.



The data displayed corresponds to a new diagnostics. When a device is selected from the list, the addressed module automatically undergoes diagnostics.

In all cases of diagnostics, the first six bytes are standardized and displayed. If a slave requires more than six bytes of diagnostics, the data is displayed in the window and can be accessed using the scroll bars.

Chapter 8
TSX PBY 100 module diagnostics

Subject of this Chapter

This chapter describes the different diagnostics functions of the TSX PBY 100 module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Diagnostics of the module's status from the LEDs	78
Degraded project modes	79
Lists of diagnostics variables	81
List of available diagnostics	83
Compact diagnostics of all slaves	84
Slave diagnostics	85
General information on a slave	86
Slave configuration data	87
Typical errors	88

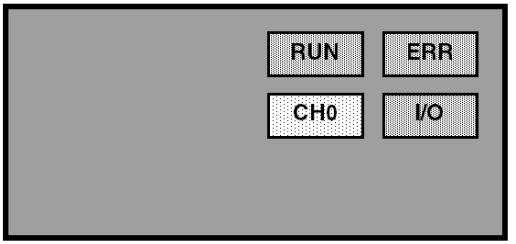
Diagnostics of the module's status from the LEDs

At a Glance

LEDs on the card are used to display the state of the module and the Profibus DP network. The signaling complies with Premium and Profibus DP standards.

Illustration

The diagnostic LEDs are as follows:



Diagnostics

Depending on the state of the LEDs, the diagnostics are as follows:

LEDs	On	Flashing	Off
RUN (green)	TSX PBY 100 module ready - self-diagnostic tests successful - Profibus DP ready	-	Module has not been initialized (awaiting configuration)
ERR (red)	Bus fault or Configuration fault or TSX PBY 100 module fault	Module awaiting configuration or Currently loading (if RUN is off) or Communication fault with the PLC (if RUN is on)	No fault indicated
I/O (red)	Fault on one or several of the slave peripherals	-	No fault indicated (all slaves are active)
CHO (yellow)	Input/output data exchange	-	No input/output data exchange

Degraded project modes

Transmission media fault

- Communication fault on starting Profibus DP:
  This fault can be caused by poor configuration or damage to the cable. In this case, the bus remains in a non-operational state and the slaves remain in failed start state.
  An error code is generated by the TSX PBY 100 master module in the form of diagnostics. All the diagnostics bits of the slaves remain in their fault state. The ERR LED is on and the other LEDs are off.
- Communication fault during operation:
  If a fault occurs while exchanges are in progress, an error code is generated by the TSX PBY 100 master module in the form of diagnostics. In this case, the slaves switch to their pre-configured default state after the watchdog period is exceeded.
  The diagnostic bits of the slaves are enabled to indicate that the slaves are not available and that inputs are reset to zero. The TSX PBY 100 module saves the diagnostics and informs the CPU of their availability using the language object %IWr.m.0.243.10...12.

TSX PBY 100 master module faults

When a fault appears, data exchanges, commands and diagnostics are interrupted. After the watchdog period is exceeded, an error code is generated in the form of diagnostics.

If exchanges are interrupted, the diagnostics bits of the slaves are enabled to indicate that the slaves are not available and that inputs are reset to zero. The ERR LED is on and the other LEDs are off.

Slave faults

When exchanges are in progress, a slave fault is indicated by a new diagnostic. If communication is still established, the slave generates the diagnostics, if not, the diagnostics are generated by the TSX PBY 100 master module.

The diagnostics bits of the slave are enabled to indicate that the slave is not available and that its inputs are reset to zero. The TSX PBY 100 module saves the diagnostics and informs the CPU of their availability using the language object %IWr.m.0.243.10...12.

NOTE: If one or several slaves are faulty, the bus cycle slows down. Several PLC cycles may be necessary for diagnostics to be recognized and inputs to be reset to zero.

General PLC CPU faults

In the event of a communication fault between the CPU and the TSX PBY 100 module, all outputs are set to their default state (maintained or Reset) and inputs are reset to zero. The ERR LED flashes to indicate the communication fault between the PLC CPU and the TSX PBY 100 module.

The transfer of diagnostics data between the master and the slave are not affected.

Resetting outputs after loading a project

For a low baud rate (less than 500 Kbit/s) and a large watchdog value, the slaves maintain their output states for the whole of the watchdog period.

For a low baud rate (less than 500 Kbit/s) and a disabled watchdog, the slave output states are maintained until the project loading has finished.

Lists of diagnostics variables

Master diagnostics

The following table indicates the diagnostics data for the TSX PBY 100 module.

Bytes	Structure	Description
0/1	OPERATING_MODE	Master operating mode (byte 0: least significant byte; byte 1: most significant byte) Hexadecimal values: 16#00 : inoperative (initialization) 16#40 : stop (ready to receive configuration) 16#80 : output fallback (depending on the configuration) 16#C0: operating
2/3	PNO_IDENTIFIER	Master identification code depending on the Profibus DPcode (byte 2 : least significant byte; byte 3: most significant byte) Hexadecimal values: 16#1654 : for the TSX PBY 100 module
4	PC card hardware version	Hardware version of the PCMCIA card depending on the Profibus DPcode Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y 16#FF: no card or card invalid
5	PC card firmware version	Software version of the PCMCIA card depending on the Profibus DPcode Hexadecimal values: 16#14 : version V5.02I 16#16 : version V5.02K 16#XY: version V5.XY 16#FF: no card or card invalid
6	PBY hardware version	Hardware version of the TSX PBY 100 module Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y
7	PBY firmware version	Software version of the TSX PBY 100 module Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y
8	PBY IE version	Index of software version for the TSX PBY 100 module Hexadecimal values: 16#04 : version IE04 16#XY: version IEXY
9...15	PC card firmware version (ASCII)	Software version of the PCMCIA card in ASCII mode

Master Class 2 diagnostics

For Profibus DP, a Master Class 2 device has the following standard TSX PBY 100 module diagnostics data:

Bytes	Structure	Description
0	OPERATING_MODE	Master operating mode Hexadecimal values: 16#00 : inoperative (initialization) 16#40 : stop (ready to receive configuration) 16#80 : output fallback (depending on the configuration) 16#C0: operating
1/2	PNO_IDENTIFIER	Master ID code depending on the Profibus DPcode (byte 1 : least significant byte; byte 2: least significant byte) Hexadecimal values: 16#1654 : for the TSX PBY 100 module
3	PC card hardware version	Hardware version of the PCMCIA card depending on the Profibus DPcode Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y 16#FF: no card or card invalid
4	PC card firmware version	Software version of the PCMCIA card depending on the Profibus DPcode Hexadecimal values: 16#14 : version V5.02I 16#16 : version V5.02K 16#XY: version V5.XY 16#FF: no card or card invalid
5	PBY hardware version	Hardware version of the TSX PBY 100 module Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y
6	PBY firmware version	Hardware version of the TSX PBY 100 module Hexadecimal values: 16#10 : version V1.0 16#XY: version VX.Y

NOTE: The most and least significant diagnostics bytes of the **PNO_IDENTIFIER** are transposed in relation to the standard diagnostics on Profibus DP.

List of available diagnostics

At a Glance

This table provides the activity bits. With one bit per slave, they provide information on the availability of new diagnostics coming from slaves.

Words	Structure	Description
0	bit 0 to 15	New diagnostics for slaves 0 to 15
1	bit 0 to 15	New diagnostics for slaves 16 to 31
2	bit 0 to 15	New diagnostics for slaves 32 to 47
3	bit 0 to 15	New diagnostics for slaves 48 to 63
4	bit 0 to 15	New diagnostics for slaves 64 to 79
5	bit 0 to 15	New diagnostics for slaves 80 to 95
6	bit 0 to 15	New diagnostics for slaves 96 to 111
7	bit 0 to 13 bit 14 and 15	New diagnostics for slaves 112 to 125 Not used

The bit is reset when the slave has been diagnosed by the SEND_REQ communication function.

All the bits can only be reset once by the SEND_REQ command, the parameter identifying this table and the parameter identifying the reset.

Compact diagnostics of all slaves

At a Glance

The diagnostics table groups together the main diagnostics for each slave. It always has a size of 125 bytes.

Each byte corresponds to a slave address, byte 0 corresponding to slave 1 and byte 124 corresponding to slave 125.

Each byte shows the same diagnostics information.

The following table describes the diagnostics information contained in each byte.

Words	Structure	Description
0...124	bit 0: NOT_REACHABLE bit 1: NOT_READY bit 2: CONFIG_FAULT bit 3: PRM_REQUIRED bit 4: INACTIVE bit 5: INVALID_RSP bit 6: PARAM_FAULT bit 7: MASTER_LOCK	bit 0=1 if slave x is not connected or switched off. bit 1=1 if slave x is not ready for data exchanges. bit 2=1 if there is a configuration error on slave x when test requested. bit 3=1 if slave x has to be reconfigured and re-parameterized. bit 4=1 if slave x is inactive (excluded from processing). bit 5=1 if there is an error in the last response from slave x. bit 6=1 if there is an error in the last parametering message from slave x. bit 7=1 if slave x has already been parameterized by another master module.

Slave diagnostics

At a Glance

Only the first six diagnostics bytes are standardized and mandatory.

One slave can provide up to 244 diagnostics bytes. With extended diagnostics (byte 7 equals ff in hexadecimal), bit 3 of byte 0 indicates this by being at 1 (bit 3=1).

Bytes	Structure	Description
0	bit 0: NOT_REACHABLE	bit 0=1 if the slave is not connected or switched off.
	bit 1: NOT_READY	bit 1=1 if the slave is not ready for data exchanges.
	bit 2: CONFIG_FAULT	bit 2=1 if there is a configuration error on the slave when test requested.
	bit 3: EXT_DIAG	bit 3=1 if there are extended diagnostics (byte 7 at FFh in hexadecimal).
	bit 4: NOT_SUPPORTED	bit 4=1 if the function is not supported by the slave.
	bit 5: INVALID_RSP	bit 5=1 if there is an error in the last response from the slave.
	bit 6: PARAM_FAULT	bit 6=1 if there is an error in the last parametering message from the slave.
	bit 7: MASTER_LOCK	bit 7=1 if the slave has already been parameterized by another master module.
1	bit 0: PRM_REQUIRED	bit 0=1 if the slave has to be reconfigured and re-parameterized.
	bit 1: DIAG_DATA_RDY	bit 1=1 if the slave has generated a diagnostics to be processed by the master.
	bit 2: IS_SLAVE_DIAG	bit 2=0 if the diagnostics has been created by the master.
		bit 2=1 if the diagnostics has been created by the slave.
	bit 3: WDT_ACTIVE	bit 3=1 if the slave watchdog is active.
	bit 4: FREEZE_MODE	bit 4=1 if the slave inputs selected are frozen.
	bit 5: SYNC_MODE	bit 5=1 if the slave outputs selected are frozen.
	bit 6:	not used.
	bit 7: INACTIVE	bit 7=1 if the slave is inactive (excluded from processing).
2	bit 0 to 6	Not used.
	bit 7: DIAG_OVERFLOW	bit 7=1 if the number of diagnostics exceeds the size of the receive words.
3	MASTER_ADDRESS	Address of the master module that sets the parameters for the slave.
4/5	PNO_IDENTIFIER	Identification code for the slave.
6...244	SPECIFIC_DIAG	Optional specific diagnostics data.

General information on a slave

At a Glance

For each slave, the following general information can be read by the TSX PBY 100 module using the SEND_REQ function.

Designation	Size	Description
Configured	byte	the slave has been configured according to Profibus DPconfiguration
Operating	byte	the slave has been initialized and is running correctly
Number of %IW	word	total size of input data in the %IW zone
Number of %QW	word	total size of output data in the %QW zone
Size of input data	byte	total size of input data on Profibus
Size of output data	byte	total size of output data on Profibus
Size of diagnostics data	byte	total size of the last received diagnostics
Compressed diagnostics	byte	compressed diagnostics data for this slave
Diagnostics counter	Byte table	total number of diagnostics messages received from the slaves, one byte per slave (the size is always 126 bytes, byte n corresponds to slave address n)
Exchange counter	Byte table	total number of communication faults between the master and his slaves, one byte per slave (the size is always 126 bytes. Byte n corresponds to slave address n)
Downtime counter	Byte table	number of times when this slave is present but unavailable (the size is always 125 bytes, byte 0 corresponds to device 1 and byte 124 to device 125)
Invalid response counter	Byte table	number of invalid responses for this slave (the size is always 125 bytes, byte 0 corresponds to device 1 and byte 124 to device 125)

Slave configuration data

At a Glance

The TSX PBY 100 module can read the configuration data from each slave with the aid of the SEND_REQ function.

Designation	Size	Description
Total length	word	total length of configuration information
%IW number	byte	total input data size in the %IW zone
%QW number	byte	total output data size in the %QW zone
Offset %IW	word	input data blocks offset in the %IW zone
Offset %QW	word	input data blocks offset in the %QW zone
Station Status	byte	refer to the Profibus DP standard
Watchdog Factor 1	byte	
Watchdog Factor 2	byte	
Min TSDR	byte	
PNO_IDENTIFIER	word	
Group Flags	byte	
ID Address	byte	slave address on the bus
Modular slave	byte	value = 1 if the slave is a modular device value = 0 if the slave is a compact device
Slave active	byte	value = 1 if slave is active on the bus value = 0 if slave is inactive on the bus
Size of parameters	word	parameter data block size for this slave
Configuration data size	word	configuration data block size for this slave
Size of data used	word	data used block size for this slave
Parameters	x bytes	parameter data block for this slave
Configuration data	x bytes	configuration data block for this slave
Data used	x bytes	block of data used for this slave

Typical errors

Case 1

ERR flashing

Results	
After loading the project, the TSX PBY 100 module's ERR LED flashes	
Causes	**Actions**
The TSX PBY 100 module is not recognized by the processor and has no configuration data.	• Test whether the Unity Pro configuration corresponds to the actual configuration, • Test whether the processor, Unity Pro and TSX PBY 100 software versions are compatible.

Case 2

ERR lit

Results	
After loading the project, the TSX PBY 100 module's ERR LED remains permanently on.	
Causes	**Actions**
Profibus DP cabling problem	• Disconnect the TSX PBY 100 module from the connection terminal and reinitialize the processor, • If the module starts correctly following this action, there is a short circuit or a wire inversion in the cabling.
Physical problem resulting from the connection terminal or PCMCIA card	• If the module does not start correctly, cut the power supply to the connection terminal and change the connection terminal and if necessary the PCMCIA card (the TAP and the card must be changed when the module is switched off).
PCMCIA card software problem	• Test the software version of the card, it must be V5.02I or above.
Problem with the loaded configuration	• Test the master module error codes and the input/output error codes, • Test the error codes via the diagnostics function.

Case 3

Line fault

Results	
After loading a project, some bus slaves start and become faulty.	
Causes	**Actions**
A line termination is detected but it is not at the end of the bus.	• Test all the Profibus DP connectors and place the bus termination at the end of the line.

Case 4

Faulty slave

Results	
A bus slave is faulty but has not caused a bus error.	
Causes	**Actions**
The slave has some input/output errors or configuration error or the watchdog is inactive	• Test the slave diagnostics data via the debugging screen.

Case 5

Delay in starting up the slave

Results	
Bus slaves do not react immediately to start-up without causing a bus error. After a while, the slave starts.	
Causes	**Actions**
Some slaves require a control command before being activated. These slaves have been too slow to react to the first command sent	• Modify the bus parameters to delay the sending of the first command. • Add 5 time units to the Timeout.

Case 6

Slaves faulty intermittently

Results	
Some slaves are intermittently faulty.	
Causes	**Actions**
Slaves are subject to cabling or electromagnetic compatibility faults, but the TSX PBY 100 module tries to reactivate them	• Reset all the diagnostics counters using the SEND_REQ function, • Test if the TSX PBY 100 module has received the new diagnostics.

Case 7

Bus hardware fault

Results	
The %IWr.m.0.243.7 bit is activated, indicating one or more bus hardware faults and all the devices transmit their diagnostics data.	
Causes	**Actions**
The bus is subject to cabling faults, connection faults, line termination faults or connection terminal faults. Because of this the slaves transmit their diagnostics and fill the receive zone with diagnostics.	• Check the cabling and line terminations, in particular the connectors whose terminations are activated but not located at the end of the bus, • Reinitialize `BUS_FLT` (%IWr.m.0.243.7) by: • switching off and then switching on the PLC, • reinitializing the PLC, • disconnecting and reconnecting the TSX PBY 10 module while switched on, • downloading a new project, • resetting all the diagnostics counters using the SEND_REQ function.

Chapter 9
Profibus DP communication language objects

Subject of this Chapter

This chapter describes the language objects associated with Profibus DP communication and the different ways of using them.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
9.1	Language objects and IODDTs for Profibus DP communication using the TSX PBY 100 module	92
9.2	General Language Objects and IODDTs for Communication Protocols	101
9.3	IODDTs for Profibus DP communication	105
9.4	The IODDT Type T_GEN_MOD Applicable to All Modules	114

Section 9.1
Language objects and IODDTs for Profibus DP communication using the TSX PBY 100 module

Subject of this Section

This section provides a general introduction to the language objects and IODDTs for Profibus DP communication using the TSX PBY 100 module

What Is in This Section?

This section contains the following topics:

Topic	Page
Presentation of the language objects for Profibus DP communication	93
Implicit Exchange Language Objects Associated with the Application-Specific Function	94
Explicit Exchange Language Objects Associated with the Application-Specific Function	95
Management of Exchanges and Reports with Explicit Objects	97

Presentation of the language objects for Profibus DP communication

General

Profibus DP communication has two associated IODDTs:

- T_COM_STS_GEN which applies to communication protocols except Fipio and Ethernet,
- T_COM_PBY which is specific to Profibus DP communication.

NOTE: IODDT variables can be created in two different ways:

- Using the **I/O objects** *(see Unity Pro, Operating Modes)* tab,
- Data Editor *(see Unity Pro, Operating Modes)*.

Language object types

In each IODDT there is a set of language objects that can be used to command them and verify their correct operation.

There are two types of language object:

- **implicit exchange objects**, which are automatically exchanged on each cycle of the task associated with the module,
- **explicit exchange objects**, which are exchanged on request by the project, using explicit exchange instructions.

Implicit exchanges concern the statuses of the modules, communication signals, slaves, etc.

Explicit exchanges allow you to set the module and perform diagnostics.

Implicit Exchange Language Objects Associated with the Application-Specific Function

At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

Reminders

The module inputs (`%I` and `%IW`) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

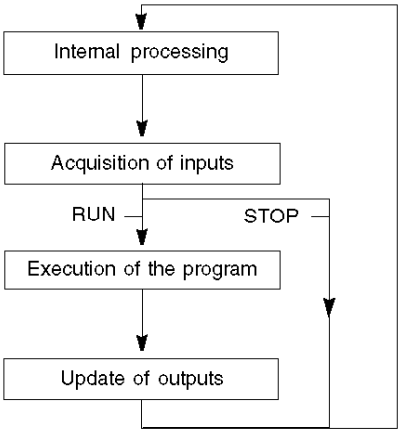
The outputs (`%Q` and `%QW`) are updated at the end of the task, only when the PLC is in RUN mode.

NOTE: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- outputs are set to fallback position (fallback mode)
- outputs are maintained at their last value (maintain mode)

Figure

The following diagram shows the operating cycle of a PLC task (cyclical execution).



Explicit Exchange Language Objects Associated with the Application-Specific Function

Introduction

Explicit exchanges are performed at the user program's request using these instructions:

- READ_STS *(see Unity Pro, I/O Management, Block Library)* (read status words)
- WRITE_CMD *(see Unity Pro, I/O Management, Block Library)* (write command words)
- WRITE_PARAM *(see Unity Pro, I/O Management, Block Library)* (write adjustment parameters)
- READ_PARAM *(see Unity Pro, I/O Management, Block Library)* (read adjustment parameters)
- SAVE_PARAM *(see Unity Pro, I/O Management, Block Library)* (save adjustment parameters)
- RESTORE_PARAM *(see Unity Pro, I/O Management, Block Library)* (restore adjustment parameters)

These exchanges apply to a set of %MW objects of the same type (status, commands or parameters) that belong to a channel.

These objects can:

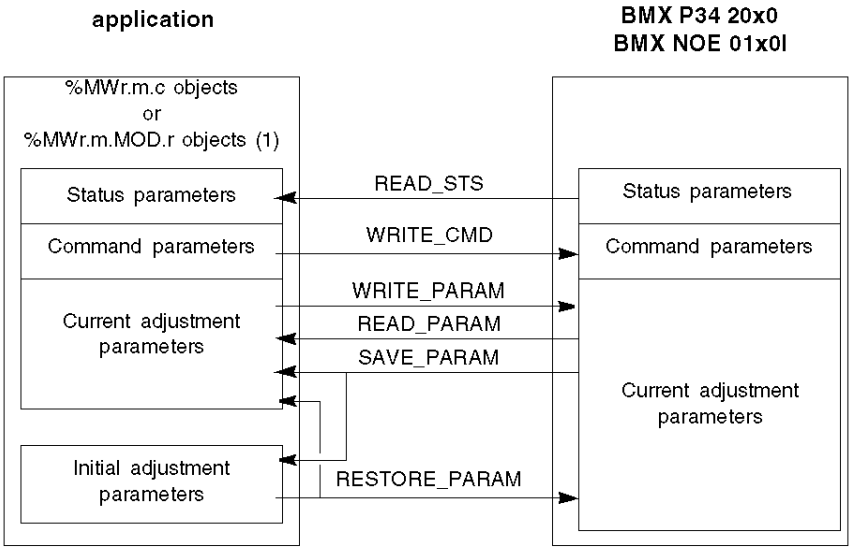
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

NOTE: To avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

NOTE: Explicit Exchanges are not supported when Modicon M340 Analog and Digital I/O modules are configured behind a M340 Ethernet Remote I/O adapter module in a Quantum EIO Ethernet Configuration. As a consequence, it is not possible to setup a module's parameters from the PLC application during operation.

General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the application and module.



(1) Only with READ_STS and WRITE_CMD instructions.

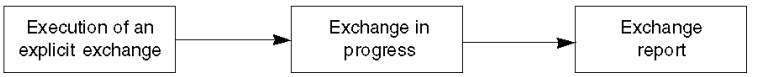
Managing Exchanges

During an explicit exchange, check performance to see that the data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress *(see page 99)*
- the exchange report *(see page 100)*

The following diagram describes the management principle for an exchange.



NOTE: In order to avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

Management of Exchanges and Reports with Explicit Objects

At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

- `EXCH_STS` (`%MWr.m.c.0`): exchange in progress
- `EXCH_RPT` (`%MWr.m.c.1`): report

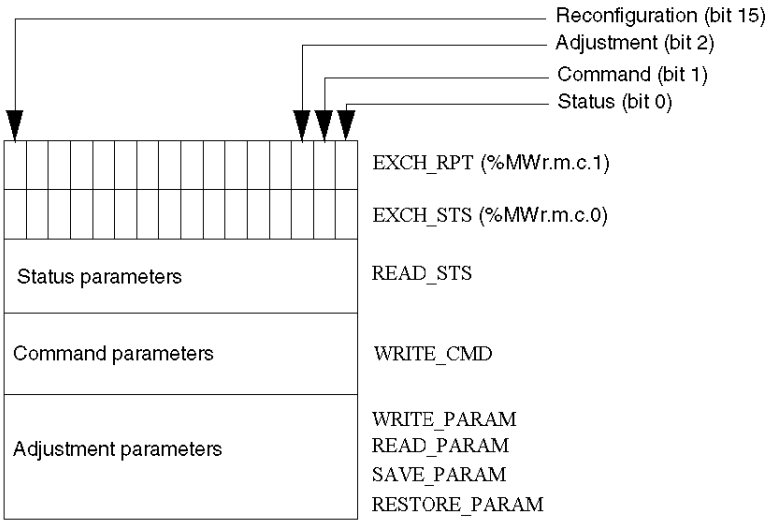
NOTE:

Depending on the localization of the module, the management of the explicit exchanges (`%MW0.0.MOD.0.0` for example) will not be detected by the application:

- For in-rack modules, explicit exchanges are done immediately on the local PLC Bus and are finished before the end of the execution task. So, the `READ_STS`, for example, is always finished when the `%MW0.0.mod.0.0` bit is checked by the application.
- For remote bus (Fipio for example), explicit exchanges are not synchronous with the execution task, so the detection is possible by the application.

Illustration

The illustration below shows the different significant bits for managing exchanges:



Description of Significant Bits

Each bit of the words EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) is associated with a type of parameter:

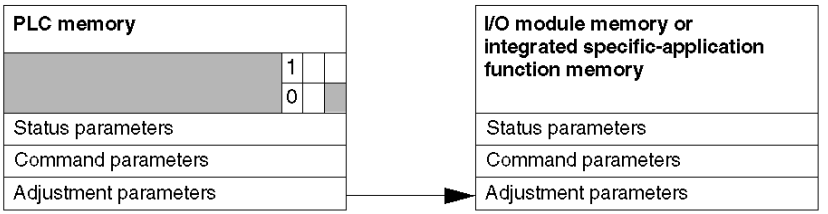
- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - The CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel.
  - The CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - The ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM).
  - The ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- Rank 15 bits indicate a reconfiguration on channel **c** of the module from the console (modification of the configuration parameters + cold start-up of the channel).
- The *r*, *m* and *c* bits indicates the following elements:
  - the **r** bit represents the rack number.
  - The **m** bit represents the position of the module in the rack.
  - The **c** bit represents the channel number in the module.

NOTE: r represents the rack number, **m** the position of the module in the rack, while **c** represents the channel number in the module.

NOTE: Exchange and report words also exist at module level EXCH_STS (%MWr.m.MOD) and EXCH_RPT (%MWr.m.MOD.1) as per IODDT type T_GEN_MOD.

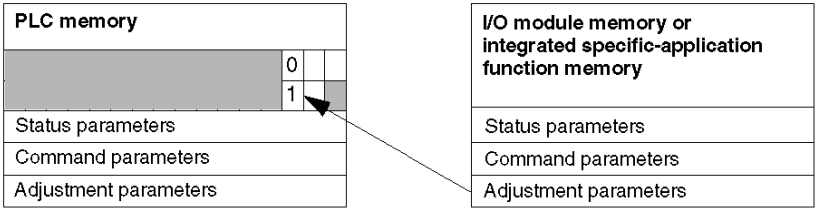
Example

Phase 1: Sending data by using the WRITE_PARAM instruction



When the instruction is scanned by the PLC processor, the **Exchange in progress** bit is set to 1 in %MWr.m.c.

Phase 2: Analysis of the data by the I/O module and report.



When the data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the ADJ_ERR bit (%MWr.m.c.1.2).

This bit makes the following reports:

- **0:** correct exchange
- **1:** faulty exchange)

NOTE: There is no adjustment parameter at module level.

Execution Indicators for an Explicit Exchange: EXCH_STS

The table below shows the control bits of the explicit exchanges: EXCH_STS (%MWr.m.c.0)

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration of the module in progress	%MWr.m.c.0.15

NOTE: If the module is not present or is disconnected, explicit exchange objects (READ_STS for example) are not sent to the module (STS_IN_PROG (%MWr.m.c.0.0) = 0), but the words are refreshed.

Explicit Exchange Report: EXCH_RPT

The table below shows the report bits: `EXCH_RPT (%MWr.m.c.1)`

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error reading channel status words (1 = failure)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during a command parameter exchange (1 = failure)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error during an adjust parameter exchange (1 = failure)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error during reconfiguration of the channel (1 = failure)	%MWr.m.c.1.15

Counting Module Use

The following table describes the steps realised between a Couting Module and the system after a power-on.

Step	Action
1	Power on.
2	The system sends the configuration parameters.
3	The system sends the adjust parameters by WRITE_PARAM method. **Note:** When the operation is finished, the bit %MWr.m.c.0.2 switches to 0.

If, in the begining of your application, you use a WRITE_PARAM command, you must wait until the bit %MWr.m.c.0.2 switches to 0.

Section 9.2
General Language Objects and IODDTs for Communication Protocols

Subject of this Section

This section presents the general language objects and IODDTs that apply to all communication protocols except Fipio and Ethernet.

What Is in This Section?

This section contains the following topics:

Topic	Page
Details of IODDT Implicit Exchange Objects of Type T_COM_STS_GEN	102
Details of IODDT Explicit Exchange Objects of Type T_COM_STS_GEN	103

Details of IODDT Implicit Exchange Objects of Type T_COM_STS_GEN

Introduction

The following table presents the IODDT implicit exchange objects of type `T_COM_STS_GEN` applicable to all communication protocols except Fipio and Ethernet.

Error Bit

The table below presents the meaning of the detected error bit `CH_ERROR` (`%Ir.m.c.ERR`).

Standard Symbol	Type	Access	Meaning	Address
CH_ERROR	EBOOL	R	Communication channel error bit.	%Ir.m.c.ERR

Details of IODDT Explicit Exchange Objects of Type T_COM_STS_GEN

Introduction

This section presents the T_COM_STS_GEN type IODDT explicit exchange objects applicable to all communication protocols except Fipio and Ethernet. It includes the word type objects whose bits have a specific meaning. These objects are presented in detail below.

Sample Variable Declaration: IODDT_VAR1 of type T_COM_STS_GEN

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Execution Flags of an Explicit Exchange: EXCH_STS

The table below shows the meaning of channel exchange control bits from channel EXCH_STS (%MWr.m.c.0).

Standard Symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Current parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The table below presents the meaning of the exchange report bits EXCH_RPT (%MWr.m.c.1).

Standard Symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Reading error for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error during adjustment parameter exchange.	%MWr.m.c.1.2

Standard Channel Faults, CH_FLT

The table below shows the meaning of the bits of the status word CH_FLT (%MWr.m.c.2). Reading is performed by a READ_STS (IODDT_VAR1).

Standard Symbol	Type	Access	Meaning	Address
NO_DEVICE	BOOL	R	No device is working on the channel.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	R	A device on the channel is inoperative.	%MWr.m.c.2.1
BLK	BOOL	R	Terminal block not connected.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Time out exceeded anomaly.	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal detected error or channel self-testing.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Interruption of the communication with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application detected error (adjustment or configuration).	%MWr.m.c.2.7

Section 9.3
IODDTs for Profibus DP communication

Subject of this Section

This section presents the language objects and IODDTs associated with Profibus DP communication using the TSX PBY 100 module

What Is in This Section?

This section contains the following topics:

Topic	Page
Details of the implicit exchange objects of the T_COM_PBY-type IODDT	106
Details of the implicit exchange language objects for a Profibus DP function	110
Language objects associated with configuration	111
Error codes for module TSX PBY 100	112

Details of the implicit exchange objects of the T_COM_PBY-type IODDT

At a Glance

The following tables present the implicit exchange objects of the T_COM_PBY-type IODDT that apply to Profibus DP communication.

Error bit

The following table presents the meaning of the CH_ERROR error bit (%Ir.m.c.ERR).

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Communication channel error bit.	%Ir.m.0.ERR

Status bits

The following table presents the meaning of the bits of status word (%IWr.m.0.242).

Standard symbol	Type	Access	Meaning	Address
CHAN_FLT	BOOL	R	bit 0 = 1: if bit 8 = 1 or bit 9 = 1 or bit 10 = 1, channel error.	%IWr.m.0.242.0
MAST_OP_FLT	BOOL	R	bit 8 = 1 Master module operating error (DP_ERROR).	%IWr.m.0.242.8
PCMCIA_OP_FLT	BOOL	R	bit 9 = 1 PCMCIA card operating error (IOM_ERROR).	%IWr.m.0.242.9
MAST_CONF_FLT	BOOL	R	bit 10 = 1 Master module configuration error (CM_ERROR).	%IWr.m.0.242.10
CONF_FLT	BOOL	R	bit 13 = 1 Configuration error.	%IWr.m.0.242.13
COM_FLT	BOOL	R	bit 14 = 1 Communication error. no communication with the programmable controller.	%IWr.m.0.242.14

Status bits

The following table presents the meaning of the bits of status word STS_243 (%IWr.m.0.243).

Standard symbol	Type	Access	Meaning	Address
-	BOOL	R	address of the last diagnosed slave.	%IWr.m.0.243.0 to 6
BUS_FLT	BOOL	R	bit 7 = 1: Hardware fault on the bus (line termination, cabling, connectors, TAP, etc).	%IWr.m.0.243.7
MAST_MOD_OP	BOOL	R	bit 8 = 1: Master module operating.	%IWr.m.0.243.8
IO_FLT	BOOL	R	bit 9 = 1: Inputs/outputs error (one or more slaves faulty).	%IWr.m.0.243.9

Standard symbol	Type	Access	Meaning	Address
NEW_MAST_DIAG	BOOL	R	bit 10 = 1: New diagnostics available for the master module.	%IWr.m.0.243.10
NEW_SLAVE_DIAG	BOOL	R	bit 11 = 1: New diagnostics available for a slave (address provided by bits 0 to 6).	%IWr.m.0.243.11
NEW_SLAVES_DIAG	BOOL	R	bit 12 = 1: New diagnostics received for several slaves.	%IWr.m.0.243.12
-	BOOL	R	Code of last management event (bus inoperative, communication error between master devices, etc.).	%IWr.m.0.243.13 to 15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.242).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_0	BOOL	R	Status bit of slave 0.	%IWr.m.0.244.0
STS_SLAVE_1	BOOL	R	Status bit of slave 1.	%IWr.m.0.244.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.244.n
STS_SLAVE_15	BOOL	R	Status bit of slave 15.	%IWr.m.0.244.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.245).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_16	BOOL	R	Status bit of slave 16.	%IWr.m.0.245.0
STS_SLAVE_17	BOOL	R	Status bit of slave 17.	%IWr.m.0.245.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.245.i
STS_SLAVE_31	BOOL	R	Status bit of slave 31.	%IWr.m.0.245.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.246).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_32	BOOL	R	Status bit of slave 32.	%IWr.m.0.246.0
STS_SLAVE_33	BOOL	R	Status bit of slave 33.	%IWr.m.0.246.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.246.i
STS_SLAVE_47	BOOL	R	Status bit of slave 47.	%IWr.m.0.246.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.247).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_48	BOOL	R	Status bit of slave 48.	%IWr.m.0.247.0
STS_SLAVE_49	BOOL	R	Status bit of slave 49.	%IWr.m.0.247.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.247.i
STS_SLAVE_63	BOOL	R	Status bit of slave 63.	%IWr.m.0.247.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.248).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_64	BOOL	R	Status bit of slave 64.	%IWr.m.0.248.0
STS_SLAVE_65	BOOL	R	Status bit of slave 65.	%IWr.m.0.248.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.248.i
STS_SLAVE_79	BOOL	R	Status bit of slave 79.	%IWr.m.0.248.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.249).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_80	BOOL	R	Status bit of slave 80.	%IWr.m.0.249.0
STS_SLAVE_81	BOOL	R	Status bit of slave 81.	%IWr.m.0.249.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.249.i
STS_SLAVE_95	BOOL	R	Status bit of slave 95.	%IWr.m.0.249.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.250).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_96	BOOL	R	Status bit of slave 96.	%IWr.m.0.250.0
STS_SLAVE_97	BOOL	R	Status bit of slave 97.	%IWr.m.0.250.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.250.i
STS_SLAVE_111	BOOL	R	Status bit of slave 111.	%IWr.m.0.250.15

Diagnostics bits

The following table presents the meaning of the bits of diagnostics word (%IWr.m.0.251).

Standard symbol	Type	Access	Meaning	Address
STS_SLAVE_112	BOOL	R	Status bit of slave 112.	%IWr.m.0.251.0
STS_SLAVE_113	BOOL	R	Status bit of slave 113.	%IWr.m.0.251.1
STS_SLAVE_n	BOOL	R	Status bit of slave n.	%IWr.m.0.251.i
STS_SLAVE_125	BOOL	R	Status bit of slave 125.	%IWr.m.0.251.13
MOD_INP_TRANS	BOOL	R	Transfer of inputs from the module to the PLC processor.	%IWr.m.0.251.14
CPU_OUTP_TRANS	BOOL	R	Transfer of outputs from the PLC processor to the module.	%IWr.m.0.251.15

Words and error codes

The following table presents the meaning of the words `MAST_ERR_COD` (%IWr.m.0.252) and `IO_ERR_COD` (%IWr.m.0.253).

Standard symbol	Type	Access	Meaning	Address
MAST_ERR_COD	INT	R	Master module error codes *(see page 112)*.	%IWr.m.0.252
IO_ERR_COD	INT	R	Inputs/outputs error codes *(see page 113)*.	%IWr.m.0.253

Details of the implicit exchange language objects for a Profibus DP function

At a Glance

The following tables present the language objects for Profibus DP communication. These objects are not integrated in the IODDTs.

List of the implicit exchange objects

The table below shows the implicit exchange objects.

Address	Type	Access	Meaning
%IWr.m.0.0 to %IWr.m.0.241	INT	R	242 DP input words.
%QWr.m.0.0 to %QWr.m.0.241	INT	R/W	242 DP output words.
%QWr.m.0.242	INT	-	%QW.r.m.0.242.0=0: Byte consistency %QW.r.m.0.242.0=1: Frame consistency
%QWr.m.0.243 to %QWr.m.0.253	INT	-	reserved.

The I/O exchange mode (%QWr.m.0.242) is available with PBY firmware V1.3IE14 or higher and requires the CPU 2.1OS version (or higher).

The application program can switch over between the following I/O exchange modes:

- I/O Exchange Mode Byte Consistency (Recommended for Discrete I/O):
  %QW.r.m.0.242.0=0 ==> Byte-by-byte data consistency within the Profibus frame, with high performance. No guarantee that all data of a Profibus frame will be consistency updated within on PLC cycle.
- I/O Exchange Mode Frame Consistency (Recommended for Analog I/O):
  %QW.r.m.0.242.0=1 ==> Data consistency over the total length of the Profibus frame, with reduced performance. In this mode the data of each Profibus frame is updated consistency within one PLC cycle.

Language objects associated with configuration

At a Glance

This page describes all configuration language objects for Profibus DP communication. These objects are not integrated in the IODDTs, but can be displayed by the application program.

Internal constants

The following table describes the internal constants:

Object	Type	Access	Meaning
%KWr.m.0.0	INT	R	bit 0 to bit 15: function code of the TSX PBY 100 module.
%KWr.m.0.1	INT	R	Number of %IW and %QW updated (32,64,128,242).
%KWr.m.0.2	INT	R	• bit 0 = 0: outputs set to zero, • bit 0 = 1: maintained outputs.

Error codes for module TSX PBY 100

Master module (MAST_ERR_COD)

Managing the internal configuration of module TSX PBY 100

Symbol	Value	Description
E_CFG_DATA_SIZE	101	Size of configuration data block invalid.
E_CFG_IO_IMAGE_SIZE	102	Size of I/O images invalid.
E_CFG_N_SLAVES	103	Number of slaves invalid.
E_CFG_MASTER_ADDRESS	104	Address of master module invalid.
E_CFG_BAUD_RATE	105	Transmission speed invalid.
E_CFG_BUS_PARAM	106	Bus parameters invalid.
E_CFG_NODE_ID	107	Address invalid or already exists.
E_CFG_SLAVE_IN_SIZE	108	Slave input data size invalid.
E_CFG_SLAVE_OUT_SIZE	109	Slave output data size invalid.
E_CFG_AAT_DATA	110	Size/offset combination of I/O data invalid.
E_CFG_AAT_OVERLAP	111	I/O data overlap.
E_CFG_CNF_TIMEOUT	112	Timeout on confirmation waiting time.
E_CFG_INIT_FMB	113	Cannot initialize PCMCIA card.
E_CFG_INIT_MASTER	114	Cannot initialize master module.
E_CFG_LOAD_BUSPAR	115	Cannot load module bus parameters.
E_CFG_SET_OPMODE	116	Cannot switch into operating mode.
E_CFG_LOAD_SLAVE	117	Cannot load slave configuration.
E_CFG_MASTER_DIAG	118	Cannot read master module diagnostics.
E_CFG_DUP_ADDR	119	Bus address already exists.
E_CFG_TAP_FAULT	120	Fault between the PCMCIA card and the TAP.

Inputs/outputs (IO_ERR_COD)

Managing the TSX PBY 100 module inputs/outputs

Symbol	Value	Description
E_OK	0	No error.
E_INIT	1	Initialization error.
E_NO_CONFIG	2	No configuration data.
E_INVALID_CONFIG	3	Invalid configuration data.
E_INVALID_PARAM	4	Invalid parameters.
E_INVALID_STATE	5	Slave state does not allow the request to be carried out.
E_ACCESS	6	No exchange on BusX.
E_NO_RESSOURCES	7	No resources available.
E_SEND	8	Cannot send message to PCMCIA card.
E_RECEIVE	9	Cannot receive message from PCMCIA card.
E_STATE	10	Invalid state.
E_SERVICE	11	Invalid service code (Uni-Telway request and facility).

Section 9.4
The IODDT Type T_GEN_MOD Applicable to All Modules

Details of the Language Objects of the T_GEN_MOD-Type IODDT

At a Glance

All the modules of Premium PLCs have an associated IODDT of type `T_GEN_MOD`.

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

List of Objects

The table below presents the objects of the IODDT:

Standard symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word.	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress.	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word.	%MWr.m.MOD.1
STS_ERR	BOOL	R	Fault when reading module status words.	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal error word of the module.	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Internal error, module failure.	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Faulty channel(s).	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block fault.	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration fault.	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative.	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal error word of the module (Fipio extension only).	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Internal fault, module unserviceable (Fipio extension only).	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Faulty channel(s) (Fipio extension only).	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block fault (Fipio extension only).	%MWr.m.MOD.2.10

Standard symbol	Type	Access	Meaning	Address
CONF_FLT_EXT	BOOL	R	Hardware or software configuration fault (Fipio extension only).	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module missing or inoperative (Fipio extension only).	%MWr.m.MOD.2.14

Index

Symbols

addressing
topological, *47*

C

channel data structure for all modules
T_GEN_MOD, *114*
channel data structure for communication protocols
T_COM_STS_GEN, *101*
channel data structure for profibus DP modules
T_COM_PBY, *105*
compliance, *37*
configuring, *51*
connecting, *32*

D

debugging, *71*
diagnosing, *78*
diagnostics, *77*, *83*

E

error codes, *112*

F

FAQs, *88*
File, *56*

I

installing, *33*

M

mapping inputs/outputs, *48*

P

parameter settings, *92*
performances, *21*
programming, *63*

T

T_COM_PBY, *105*
T_GEN_MOD, *114*
topologies, *15*
TSXPBY100, *27*